UIUCDCS-R-75-758

AN ITERATIVE METHOD FOR SOLVING NONSYMMETRIC LINEAR SYSTEMS WITH DYNAMIC ESTIMATION OF PARAMETERS

by

Thomas Albert Manteuffel

October 1975

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

UIUCDCS-R-75-758

AN ITERATIVE METHOD FOR SOLVING NONSYMMETRIC LINEAR SYSTEMS WITH DYNAMIC ESTIMATION OF PARAMETERS

by

Thomas Albert Manteuffel

October 1975

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

This work was supported in part by the Department of Computer Science and in part by the National Science Foundation under grants GJ-36393 and DCR74-23679, and was submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Mathematics, 1975.

ACKNOWLEDGMENT

I would like to acknowledge the guidance and encouragement of my advisor, Professor Paul E. Saylor. His support, both financial and personal, made this thesis possible. I am grateful to the National Science Foundation for their support under grants NSF GJ-36393 and DCR74-23679 (NSF).

I would also like to thank Connie Slovak for her excellent typing and Mr. Stanley Zundo for preparing the drawings that appear in this thesis.

Finally, I would like to acknowledge the support and encouragement of Mary Stoddard Manteuffel throughout my graduate studies.

TABLE OF CONTENTS

1. INTRODUCTION AND PRELIMINARIES

1.1 Introduction

In applications such as reservoir problems, reactor studies, and numerical weather forecasting, one often encounters partial differential equation boundary value problems. A standard technique for solving these problems is to approximate the solution of the differential equation at a discrete set of points by the solution of a linear system, $Ax = b$. In general, such systems are large, sparse, and often nonsymmetric. In this thesis I will discuss an iterative method for solving large sparse nonsymmetric linear systems.

In applications, systems arising from partial differential equations may be quite large, up to 1,000,000 unknowns, with as few as 5 to 10 nonzero entries per equation. The size and sparsity of these systems motivate the use of an iterative method. Direct methods require a prohibitive amount of storage.

The iteration to be discussed is a polynomial based gradient method (Rutishauser [7]) based on the scaled and translated Tchebychef polynomials. In the following I will show that when applied to systems whose spectrum lies in the right half complex plane, the Tchebychef iteration is optimal, in a certain sense, over all polynomial based gradient methods. Further, I will show how the optimal iteration parameters associated with the Tchebychef iteration can be computed from knowledge of the spectrum of the system.

A drawback to most iterative methods is that the choice of optimal parameters depends upon prior knowledge of the eigenvalue structure of the system. In this thesis I will develop an adaptive procedure for determining the spectrum of the matrix during execution at a relatively low cost. Optimal iteration parameters can be computed dynamically with little prior knowledge of the spectrum.

The adpative Tchebychef algorithm has the following properties:

1. Little a priori knowledge of the spectrum is required.
2. The method does not depend upon any special structure of the matrix A.
3. The method is sensitive to the condition of the matrix A, rather than the matrix $A^T A$.
4. The method requires only one matrix vector multiplication per iterative step.
5. The method can be used in conjunction with factorization methods.
6. The method may be used on singular systems.

Few iterative methods have been developed to treat nonsymmetric systems. None share these properties with the adaptive Tchebychef algorithm. In a variety of test problems, the Tchebychef method converged considerably faster than two competing methods.

Some work on nonsymmetric systems has been done by Wrigley [25], Kjellberg [16], Kincaid [15], and Faddeev [8]. This work is an extension of work done by Diamond [3] and Wrigley [25]. Many of the ideas for this thesis came from work done by Hestenes [12], Rutishauser [7], Stiefel [7], [12], [19], Golub [10], and Varga [10], [21].

The main ideas of this thesis are as follows: Chapter 1 presents the background of polynomial based gradient methods. Section 1.2 describes the type of system to be considered. A crude region containing the spectrum of the system is determined. Section 1.3 outlines the general polynomial based gradient method and establishes criteria for choosing a sequence of polynomials upon which to base an iterative method.

Chapter 2 deals with the scaled and translated Tchebychef polynomials in the complex plane. Section 2.1 describes the properties of the scaled and translated Tchebychef polynomials. It is shown that the asymptotic behavior of the Tchebychef polynomials is related to ellipses in the complex plane. Sections 2.2, 2.3, and 2.4 show that the scaled and translated Tchebychef polynomials fit the criteria established in Section 1.3. The iteration based on the Tchebychef polynomials is described in Section 2.4 where it is shown that the iteration parameters can be determined from the scaling and translating parameters c and d.

The problem of choosing the parameters c and d is outlined in Chapter 3. In Section 3.1 a measure of the rate of convergence is established at each eigenvalue as a function of the parameters c and d. A mini-max problem is constructed in terms of the eigenvalues and the parameters c and d whose solution represents the "best" choice of parameters c and d. The mini-max problem is refined and reformulated in Section 3.2. In Section 3.3 it is shown that for a real matrix A the iteration can be carried out in real arithmetic.

Chapter 4 is devoted to finding the solution to the mini-max problem constructed in Chapter 3. Section 4.1 shows that the solution of the mini-max problem can be found in terms of the solution of a mini-max problem when the spectrum of the matrix A contains 1, 2, or 3 complex conjugate pairs of eigenvalues. The remainder of the chapter is devoted to solving the mini-max problem in these special cases. Section 4.4 contains the algorithm for solving the mini-max problem.

Chapter 5 describes a method of estimating the eigenvalue structure of the matrix A during execution and a dynamic procedure for improving the choice of parameters c and d. Section 5.1 deals with an adaptation of the modified power method for simultaneous estimation of several eigenvalues based on the sequence of residuals (Wilkinson [24]). Section 5.2 shows how each new eigenvalue estimate can be added to previous estimates to yield an improved choice of parameters c and d.

Chapter 6 gives a discussion of the implementation of the algorithm followed by a discussion of competing methods and experimental results. It is shown that the adaptive Tchebychef algorithm requires fewer iterative steps to produce convergence and half as much work per step than two competing methods: Bidiagonalization (Golub and Kahan [11]) and the method of Conjugate Gradients (Hesteries and Stiefel [12]) applied to the equivalent system $A^TAx = A^Tb$.

The appendix contains a listing of the algorithm, coded in FORTRAN.

1.2 Type of System

The adaptive Tchebychef algorithm is restricted to systems with eigenvalues in the right half (left half) complex plane. Such systems arise naturally from applications to partial differential equations, especially in conjunction with finite element methods (Zienkiewicz [27]).

In the remainder of this thesis the matrix A will be a real valued matrix with eigenvalues, λ_i, in the open right half plane. The possibility of zero eigenvalues will be discussed in Chapter 6.

To establish bounds on the spectrum of such a matrix, let $M = \frac{A+A^T}{2}$ and $N = \frac{A-A^T}{2}$. It is clear that M is symmetric, N is antisymmetric, and $A = M + N$. Suppose λ is an eigenvalue of A corresponding to the eigenvector $\nu = x + iy$. Then,

$$\lambda = \frac{\langle A\nu,\nu\rangle}{\langle\nu,\nu\rangle} = \frac{\langle M\nu,\nu\rangle}{\langle\nu,\nu\rangle} + \frac{\langle N\nu,\nu\rangle}{\langle\nu,\nu\rangle} .$$

Because M is symmetric, we have

$$\begin{aligned}\langle M\nu,\nu\rangle &= \langle M(x+iy),\ (x+iy)\rangle = \langle Mx,x\rangle + i\langle My,x\rangle - i\langle Mx,y\rangle + \langle My,y\rangle \\ &= \langle Mx,x\rangle + \langle My,y\rangle.\end{aligned}$$

Since N is antisymmetric, $\langle Nx,x\rangle = \langle Ny,y\rangle = 0$, and we have

$$\begin{aligned}\langle N\nu,\nu\rangle &= \langle N(x+iy),\ (x+iy)\rangle = \langle Nx,x\rangle + i\langle Ny,x\rangle - i\langle Nx,y\rangle + \langle Ny,y\rangle \\ &= 2i\langle Ny,x\rangle.\end{aligned}$$

Thus, we have

$$\mathrm{Re}(\lambda) = \frac{\langle Mx,x\rangle + \langle My,y\rangle}{\|x\|^2 + \|y\|^2} ,$$

and

$$|\mathrm{Im}(\lambda)| = \frac{2|\langle Ny,x\rangle|}{\|x\|^2 + \|y\|^2} \leq \frac{2\|N\| \cdot \|x\| \cdot \|y\|}{\|x\|^2 + \|y\|^2} \leq \|N\| \, .$$

Since M is symmetric, its spectrum is contained in some interval, [a,b], on the real line. Since N is antisymmetric, its spectrum lies along the imaginary axis and $\|N\|$ is equal to the spectral radius of N. If λ is an eigenvalue of A, then

$$\mathrm{Re}(\lambda) \in [a\ b] \, ,$$
$$|\mathrm{Im}(\lambda)| \leq \|N\| \, .$$

Using Gershgorin's circle theorem, bounds for the interval [a,b] and $\|N\|$ can be found in terms of the elements of the matrices M and N (Varga [22]) giving a rectangle known to contain the spectrum of the operator A (see Figure 1.1 and Figure 1.2).

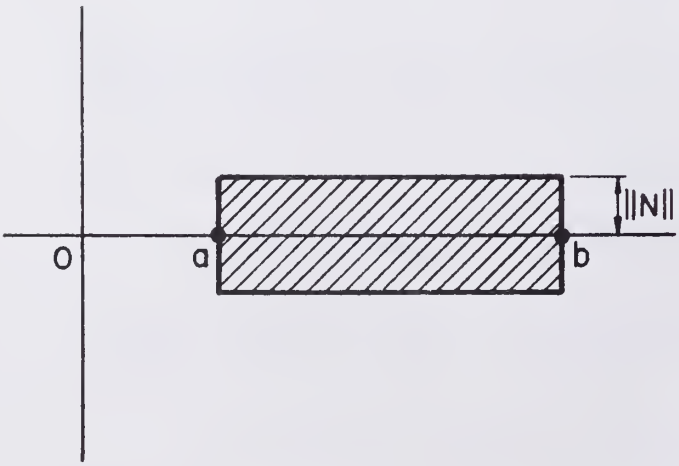

Figure 1.1

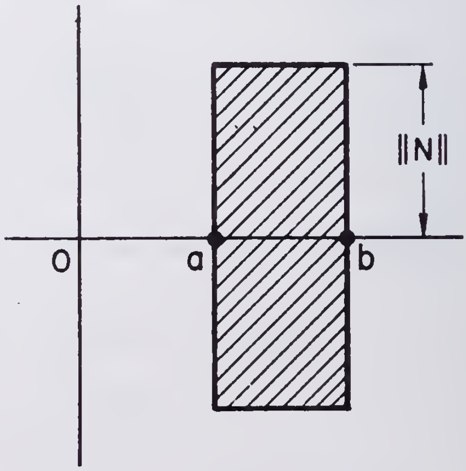

Figure 1.2

Notice that if $\|N\|$ is small with respect to $|b-a|$, the rectangle is like that shown in Figure 1.1. If $\|N\|$ is large with respect to $|b-a|$, then the rectangle is like that shown in Figure 1.2. Both rectangles are more closely approximated by an ellipse than a circle. The importance of this observation will be shown later.

If M is positive definite, then the spectrum of A lies in the right half plane. In particular, if A is diagonally dominant with positive real diagonal elements, then M will be positive definite. This type of matrix is often encountered in application. Of particular interest are positive definite systems perturbed by a small nonsymmetric part, yielding a system with its spectrum contained in a region like that shown in Figure 1.1. It is this type of system for which the Tchebychef algorithm has the greatest advantage over competing methods.

1.3 Type of Iteration

Suppose we want to solve the system $Ax = b$. If x_0 is an initial guess at the solution, x, we can compute the initial residual

$$r_0 = A(x-x_0) = b - Ax_0 ,$$

and use this information to make a new guess, x_1. Again we can compute the residual

$$r_1 = A(x-x_1) = b - Ax_1 .$$

Utilizing all of the information at hand we can use both r_0 and r_1 to make a new guess, x_2. This leads to an iteration with general step

$$x_{n+1} = x_n + \sum_{i=1}^{n} \gamma_{ni} r_i ,$$

where the γ_{ij}'s are constants. At each step we add a linear combination of the previous residuals. Let $e_n = x - x_n$ be the error at the n^{th} step. From the general step we can write

$$e_{n+1} = e_n - \sum_{i=1}^{n} \gamma_{ni} r_i = e_n - A \sum_{i=1}^{n} \gamma_{ni} e_i .$$

Lemma 1.1 The error at the n^{th} step is given by

$$e_n = P_n(A) e_0 ,$$

where $P_n(z)$ is a polynomial of degree n such that $P_n(0) = 1$.

Proof The proof will proceed by induction. For n = 1 we have

$$e_1 = e_0 - \gamma_{00} A e_0 = (I - \gamma_{00} A) e_0 ,$$

so that we may define

$$P_1(z) = (1 - \gamma_{00} z) ,$$

and we have $P_1(0) = 1$.

Assume the conclusion is true for $i \leq n$; then,

$$\begin{aligned} e_{n+1} &= e_n - A \sum_{i=1}^{n} \gamma_{ni} e_i \\ &= e_n - \gamma_{nn} A e_n - A \sum_{i=1}^{n-1} \gamma_{ni} e_i \\ &= [(I - \gamma_{nn} A) P_n(A) - A \sum_{i=1}^{n-1} \gamma_{ni} P_i(A)] e_0 . \end{aligned}$$

Let

$$P_{n+1}(z) = (1 - \gamma_{nn} z)P_n(z) - z \sum_{i=1}^{n-1} \gamma_{ni} P_i(z) \quad .$$

Then, $P_{n+1}(z)$ is a polynomial of degree n+1 and $P_{n+1}(0) = 1$. This proves the lemma.

Any sequence of polynomials, $\{P_n(z)\}$, can be generated by choosing the constants, $\{\gamma_{ij}\}$. We would like to choose the sequence of polynomials so that

$$\|e_n\| \leq \|P_n(A)\| \; \|e_0\|$$

is small. Let us examine $P_n(A)$. If A is diagonalizable, then the Jordan form of A is a diagonal matrix (Birkhoff and MacLane [1]). There exists a nonsingular S such that $A = S^{-1}JS$ and

$$J = \begin{pmatrix} \lambda_1 & & & \\ & \cdot & & \\ & & \cdot & \\ & & & \lambda_k \end{pmatrix}.$$

Now

$$P_n(A) = P_n(S^{-1}JS) = S^{-1}P_n(J)S \; ,$$

and since J is a diagonal matrix we have

$$P_n(J) = \begin{pmatrix} P_n(\lambda_1) & & & \\ & \cdot & & \\ & & \cdot & \\ & & & P_n(\lambda_k) \end{pmatrix}.$$

This yields the following result:

<u>Theorem 1.2</u> If A is diagonalizable, then $\|P_n(A)\| \to 0$ as $n \to \infty$ if and only if $P_n(\lambda_i) \to 0$ as $n \to \infty$ for every eigenvalue, λ_i, of A.

<u>Proof</u> The proof is clear from the discussion above.

Suppose A is not diagonalizable; that is, suppose A has nonlinear elementary divisors. The Jordan form of A has nontrivial Jordan blocks. We have $A = S^{-1}JS$ where

$$J = \begin{pmatrix} J_1 & & \\ & \ddots & \\ & & J_k \end{pmatrix},$$

and

$$J_i = \begin{pmatrix} \lambda_i & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{pmatrix}$$

is the Jordan block associated with the eigenvalue λ_i of multiplicity d_i. In this case

$$P_n(A) = P_n(S^{-1}JS) = S^{-1}P_n(J)S,$$

and

$$P_n(J) = \begin{pmatrix} P_n(J_1) & & \\ & \ddots & \\ & & P_n(J_k) \end{pmatrix}.$$

Taking successive powers of J_i we see that

$$J_i^2 = \begin{pmatrix} \lambda_i^2 & 2\lambda_i & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & \ddots & 2\lambda_i \\ & & & & \lambda_i^2 \end{pmatrix},$$

$$J_i^3 = \begin{pmatrix} \lambda_i^3 & 3\lambda_i^2 & 3\lambda_i & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots & 1 \\ & & & \ddots & \ddots & 3\lambda_i \\ & & & & \ddots & 3\lambda_i^2 \\ & & & & & \lambda_i^3 \end{pmatrix},$$

and in general,

$$J_i^s = \begin{pmatrix} \lambda_i^s & \binom{s}{1}\lambda_i^{s-1} & \binom{s}{2}\lambda_i^{s-2} & \cdots & \binom{s}{d_i}\lambda_i^{s-d_i+1} \\ & \ddots & \ddots & \ddots & \vdots \\ & & \ddots & \ddots & \binom{s}{2}\lambda_i^{s-2} \\ & & & \ddots & \binom{s}{1}\lambda_i^{s-1} \\ & & & & \lambda_i^s \end{pmatrix}$$

$$= \begin{pmatrix} \lambda_i^s & \frac{s}{1}\lambda_i^{s-1} & \frac{s(s-1)}{2!}\lambda_i^{s-2} & \cdots & \frac{s(s-1)\ldots(s-r+1)}{(d_i-1)!}\lambda_i^{s-d_i+1} \\ & \ddots & \ddots & \ddots & \vdots \\ & & \ddots & \ddots & \frac{s(s-1)}{2!}\lambda_i^{s-2} \\ & & & \ddots & \frac{s}{1}\lambda_i^{s-1} \\ & & & & \lambda_i^s \end{pmatrix}.$$

The superdiagonal terms act like derivatives. Since $P_n(J_i)$ is a linear combination of powers of J_i, we have

$$P_n(J_i) = \begin{pmatrix} P_n(\lambda_i) & P_n'(\lambda_i) & \frac{1}{2!}P_n''(\lambda_i) & \cdots & \frac{1}{(d_i-1)!}P_n^{(d_i-1)}(\lambda_i) \\ & \ddots & \ddots & \ddots & \vdots \\ & & \ddots & \ddots & \frac{1}{2!}P_n''(\lambda_i) \\ & & & \ddots & P_n'(\lambda_i) \\ & & & & P_n(\lambda_i) \end{pmatrix}$$

This yields the following theorem:

Theorem 1.3 If λ_i is an eigenvalue of A with multiplicity d_i, then $\|P_n(A)\| \to 0$ as $n \to \infty$ if and only if $P_n^{(j)}(\lambda_i) \to 0$ as $n \to \infty$ for every $j < d_i$, for each eigenvalue λ_i.

Proof The proof is clear from the discussion above.

When looking for a sequence of polynomials to suppress the eigenvalues of A, we must find one whose derivatives also suppress the eigenvalues of A.

In light of the previous discussion, we can establish three criteria upon which to choose a sequence of polynomials:

1. We must choose $P_n(z)$, among polynomials of like degree such that $P_n(0) = 1$, to be "as small as possible" on the spectrum of A.
2. If A has nonlinear elementary divisers, then we must choose $P_n(z)$ so that its derivatives are small on the spectrum of A.

3. We must choose $P_n(z)$ to have some recursive properties so that all of the previous residuals need not be stored.

In the next chapter we will see that the scaled and translated Tchebychef polynomials fit these criteria.

2. TCHEBYCHEF ITERATION IN THE COMPLEX PLANE

The Tchebychef polynomials were discovered a century ago by the Russian mathematician Tchebychef (the spelling of which has many variations). Their importance for practical computation, however, was rediscovered about thirty years ago by C. Lanczos. Since then they have found many uses in numerical analysis (Fox [9]).

The definition and basic properties of the Tchebychef polynomials in the complex plane will be discussed in Section 2.1. Sections 2.2, 2.3, and 2.4 will show how the Tchebychef polynomials meet the criteria established in Section 1.3. The gradient method based on the Tchebychef polynomials is developed in Section 2.4.

2.1 The Tchebychef Polynomials

The Tchebychef polynomials are given by:

$$T_0(z) = 1,$$

$$T_1(z) = z,$$

$$T_{n+1}(z) = 2zT_n(z) - T_{n-1}(z), \quad n > 1 .$$

They may also be written:

$$T_n(z) = \cosh(n \cosh^{-1}(z)),$$

where the branch of $\cosh^{-1}$ with positive real part is used.

Consider the map $\eta = \cosh(z)$. Let $z = x + iy$, $\eta = u + iv$. Then $\cosh(z) = \cosh(x+iy) = u + iv = \eta$. Using the expansion formula for the cosh, we have

$$\cosh(x+iy) = \cosh(x) \cos(y) + i \sinh(x) \sin(y),$$

or

$$u = \cosh(x) \cos(y),$$

$$v = \sinh(x) \sin(y).$$

Suppose we fix $x > 0$ and allow y to vary. Then u and v satisfy

$$\frac{u^2}{\cosh^2(x)} + \frac{v^2}{\sinh^2(x)} = 1 .$$

That is, the line x = constant is mapped onto an ellipse with semi-major axis $|\cosh(x)|$, semi-minor axis $|\sinh(x)|$, and foci at 1 and -1 (see Figure 2.1). This map has period $2\pi i$. Since $\cosh(x)$ and $\sinh(x)$ are increasing for $x \geq 0$, if $0 < a < b$, then the line $x = a$ is mapped onto an ellipse inside and confocal to the ellipse that the line $x = b$ is mapped onto (see Figure 2.1). If $x = 0$ we have

$$u = \cos(y),$$

$$v = 0 ,$$

and the ellipse has collapsed onto the real line segment $[-1, 1]$.

Because of periodicity, the map $\eta = \cosh(z)$ takes the region shown in Figure 2.2 onto the entire η-plane. Each vertical line in this region is mapped onto an ellipse in the η-plane. This region is the branch of $\cosh^{-1}$ used in the definition of the Tchebychef polynomials.

The function $\cosh^{-1}$ may also be written in log form:

$$\cosh^{-1}(w) = \ln(w + (w^2-1)^{\frac{1}{2}}) .$$

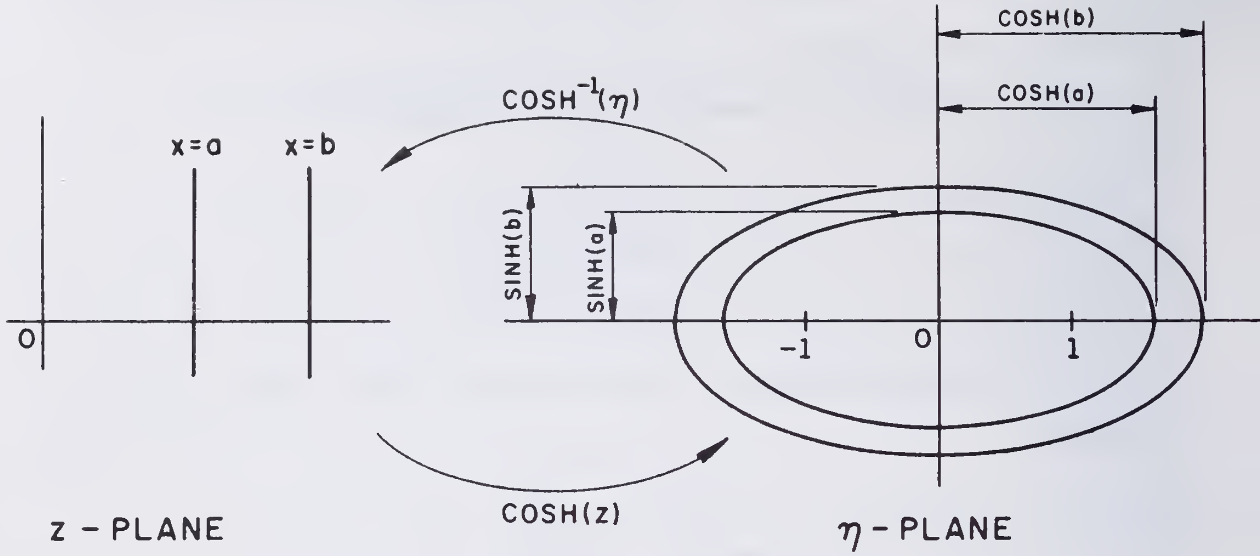

Figure 2.1

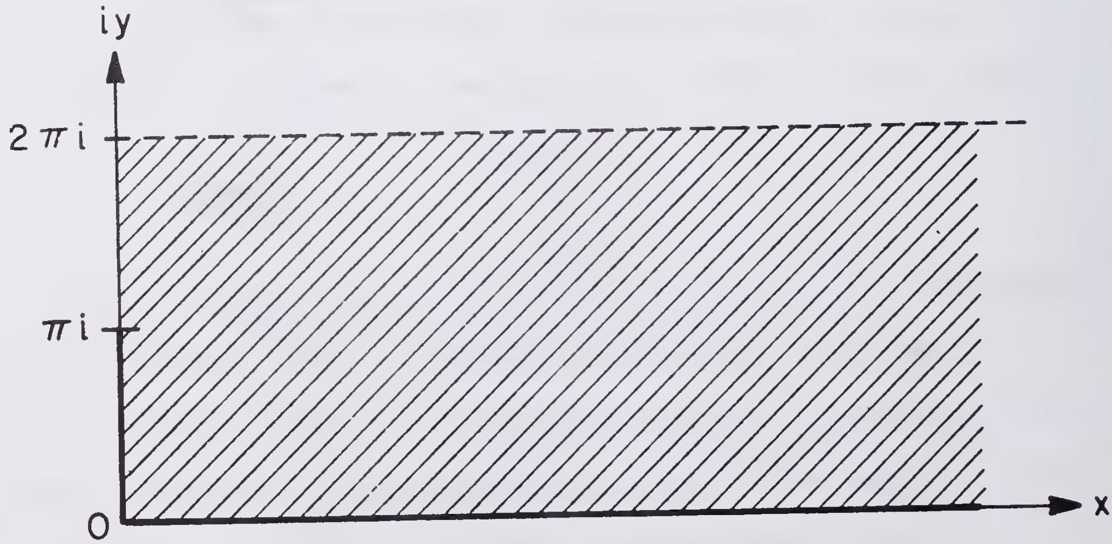

Figure 2.2

Care must be taken when choosing the branch of the square root. The branch chosen depends on the argument w and should be chosen so that $(w^2)^{\frac{1}{2}} = w$.

The n^{th} Tchebychef polynomial, $T_n(z) = \cosh(n \cosh^{-1}(z))$, maps an ellipse onto a vertical line segment in the region shown in Figure 2.2, multiplies this line segment by n, and maps the new line segment back onto another ellipse (see Figure 2.1). Since the new line segment cuts through n branches of $\cosh^{-1}$, it is wrapped around the new ellipse n times. The degenerate ellipse, the line segment [-1, 1], is mapped onto the line segment $[0, i\pi]$, multiplied by n, and mapped back onto the line segment [-1, 1]. Since it is wrapped around n times, $T_n(z)$ has n zeros on the line segment [-1, 1].

To establish some notation, let $\mathcal{F}(d,c)$ be the family of ellipses centered at d with foci at d+c and d-c. Let $F(d,c) \in \mathcal{F}(d,c)$ be a member of this family. Let $F_i(d,c) \subset F_j(d,c)$ mean that the ellipse $F_i(d,c)$ is inside the ellipse $F_j(d,c)$. Let $z \in F_i(d,c)$ mean the point z is on the ellipse $F_i(d,c)$. The Tchebychef polynomials then map members of $\mathcal{F}(0,1)$ onto other members of $\mathcal{F}(0,1)$.

Lemma 2.1 Suppose $z_i \in F_i(0,1)$, $z_j \in F_j(0,1)$; then,

$$\mathrm{Re}(\cosh^{-1}(z_i)) < \mathrm{Re}(\cosh^{-1}(z_j)) \Leftrightarrow F_i(0,1) \subset F_j(0,1),$$

$$\mathrm{Re}(\cosh^{-1}(z_i)) = \mathrm{Re}(\cosh^{-1}(z_j)) \Leftrightarrow F_i(0,1) = F_j(0,1).$$

Proof The proof follows from the discussion above.

Now consider the scaled Tchebychef polynomials,

$$C_n(z) = \frac{T_n(z)}{T_n(z_0)}, \quad z_0 \notin [-1, 1] .$$

These polynomials exhibit an asymptotic behavior.

<u>Lemma 2.2</u> If $C_n(z) = \frac{T_n(z)}{T_n(z_0)}$, $z_0 \notin [-1, 1]$, then for large n

$$C_n(z) \doteq e^{n[\cosh^{-1}(z) - \cosh^{-1}(z_0)]} .$$

<u>Proof</u> From the definition of the cosh, we have

$$C_n(z) = \frac{e^{n \cosh^{-1}(z)} + e^{-n \cosh^{-1}(z)}}{e^{n \cosh^{-1}(z_0)} + e^{-n \cosh^{-1}(z_0)}} .$$

Since the branch of $\cosh^{-1}$ with positive real part is used the result follows.

For large n, $C_n(z)$ takes on the form r^n, which motivates the following definition.

<u>Definition</u> Let

$$r(z) = \lim_{z \to \infty} |C_n(z)^{\frac{1}{n}}|$$

be the <u>asymptotic convergence factor</u> of $C_n(z)$ at the point z.

The asymptotic convergence factor, which will be referred to as the convergence factor, is related to the rate of convergence as defined by Young [26] and Varga [22] in that rate of convergence equals $-\ln(r(z))$. From the lemma above we have

$$r(z) = |e^{\cosh^{-1}(z) - \cosh^{-1}(z_0)}| = e^{\mathrm{Re}(\cosh^{-1}(z)) - \mathrm{Re}(\cosh^{-1}(z_0))}$$

There is a relationship between the convergence factor and the members of $\mathcal{F}(0,1)$, the family of ellipses with foci at 1 and -1.

<u>Lemma 2.3</u> If $z_0 \in F_0(0,1)$, $z_i \in F_i(0,1)$, $z_j \in F_j(0,1)$, then

$$r(z_i) < r(z_j) \Leftrightarrow F_i(0,1) \subset F_j(0,1),$$

$$r(z_i) = r(z_j) \leftrightarrow F_i(0,1) = F_j(0,1),$$
$$r(z) = 1 \leftrightarrow z \in F_0(0,1) \ .$$

Proof Since $r(z) = e^{\mathrm{Re}(\cosh^{-1}(z)) - \mathrm{Re}(\cosh^{-1}(z_0))}$, the result follows from the previous lemmas.

Theorem 2.4 Let $C_n(z) = \frac{T_n(z)}{T_n(z_0)}$, $z_0 \notin [-1, 1]$. If $F_0(0,1)$ is the member of the family $\mathcal{F}(0,1)$ passing through z_0, then

$$\lim_{n\to\infty} C_n(z) = \begin{cases} 0 & \text{if } z \text{ is inside } F_0(0,1) \\ \infty & \text{if } z \text{ is outside } F_0(0,1) \ . \end{cases}$$

Proof From Lemma 2.3 we see that $r(z) < 1$ if z is inside $F_0(0,1)$ and $r(z) > 1$ if z is outside $F_0(0,1)$. Since $|C_n(z)| \doteq r(z)^n$ the result follows.

Consider the transformation $z = \frac{d-\lambda}{c}$, $z_0 = \frac{d}{c}$, where d and c are any complex numbers. Let

$$P_n(\lambda) = C_n(z) = \frac{T_n(\frac{d-\lambda}{c})}{T_n(\frac{d}{c})} \ .$$

The polynomials $P_n(\lambda)$ are called the scaled and translated Tchebychef polynomials. Notice that $P_n(0) = 1$ as is required for use in a gradient method.

The choice of d and c determines a family of ellipses, $\mathcal{F}(d,c)$, with foci at d+c and d-c (see Figure 2.3). Let $F_0(d,c) \in \mathcal{F}(d,c)$ be the member of the family passing through the origin. The transformation maps members of $\mathcal{F}(d,c)$ in the λ-plane onto members of $\mathcal{F}(0,1)$ in the z-plane.

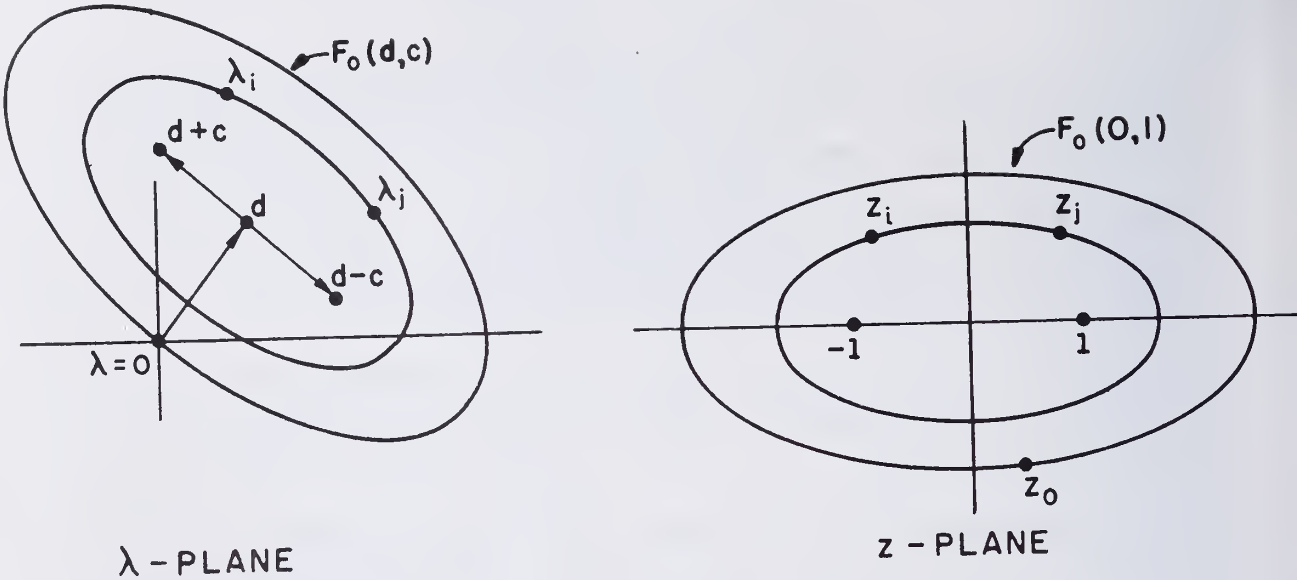

Figure 2.3

As before, let

$$r(\lambda) = \lim_{n\to\infty} |P_n(\lambda)^{\frac{1}{n}}|$$

be the asymptotic convergence factor of $P_n(\lambda)$ at the point λ. We have from above

$$r(\lambda) = e^{\mathrm{Re}(\cosh^{-1}(\frac{d-\lambda}{c})) - \mathrm{Re}(\cosh^{-1}(\frac{d}{c}))} .$$

The relationship between the members of $\mathcal{F}(d,c)$ and the convergence properties of the polynomials $P_n(\lambda)$ is given in Theorem 2.5.

<u>Theorem 2.5</u> If $P_n(\lambda) = \dfrac{T_n(\frac{d-\lambda}{c})}{T_n(\frac{d}{c})}$, then

1.
$$\lim_{n\to\infty} P_n(\lambda) = \begin{cases} 0 & \text{if } \lambda \text{ is inside } F_0(d,c) \\ \infty & \text{if } \lambda \text{ is outside } F_0(d,c) \end{cases} .$$

If $\lambda_i \in F_i(d,c)$, $\lambda_j \in F_j(d,c)$, then

2.
$$r(\lambda_i) < r(\lambda_j) \leftrightarrow F_i(d,c) \subset F_j(d,c),$$
$$r(\lambda_i) = r(\lambda_j) \leftrightarrow F_i(d,c) = F_j(d,c),$$
$$r(\lambda) \;= 1 \qquad \leftrightarrow \lambda \in F_0(d,c) \;.$$

Proof Since the transformation maps members of $\mathcal{F}(d,c)$ in the λ-plane onto members of $\mathcal{F}(0,1)$ in the z-plane, the result follows from Theorem 2.4 and Lemma 2.3.

2.2 Optimal Properties of the Tchebychef Polynomials

The first criterion mentioned in Section 1.3 suggests that when choosing a sequence of polynomials upon which to base an iterative method, it is desirable to choose polynomials that are small on the spectrum of the matrix A. Since the spectrum of A is seldom known, it is more practical to choose polynomials that are small on a region containing the spectrum of A. If the region is bounded by a circle or an ellipse, the scaled and translated Tchebychef polynomials have certain optimal properties. Much is known of the optimal properties of the monic scaled and translated Tchebychef polynomials over all monic polynomials of like degree on regions bounded by ellipses (Hille [13], Walsh [23]). Similar results, but not as strong, can be shown for polynomials normalized at the origin.

Definition Let S_n = {all polynomials, $s_n(\lambda)$, of degree n such that $s_n(0) = 1$}. The elements of S_n are said to be normalized at the origin.

Theorem 2.6 Let E be a closed and bounded infinite set in the complex plane. There exists a unique $t_n \in S_n$ such that

$$\max_{z\in E} |t_n(z)| = \min_{s_n \in S_n} \max_{z\in E} |s_n(z)|.$$

Proof Omitted (Hille [13]).

If the region E is bounded by an ellipse, a circle being a special case of an ellipse, then the maximum will occur on the boundary. Using the notation of Section 2.1, let $F_a(d,c)$ be the member of the family $\mathcal{F}(d,c)$ with semi-major axis $a < 0$. Instead of taking the maximum over the entire region we may take the maximum over the boundary, $F_a(d,c)$. The circle with center d and radius a is denoted $F_a(d,0)$. If the spectrum of A is contained in a region that is bounded by a circle that does not include the origin in its interior, we have the following result.

Theorem 2.7 Suppose $F_a(d,0)$ does not include the origin in its interior. Let $a \leq |d|$. The unique polynomial $t_n \in S_n$ such that

$$\max_{\lambda\in F_a(d,0)} |t_n(\lambda)| = \min_{s_n\in S_n} \max_{\lambda\in F_a(d,0)} |s_n(\lambda)|$$

is given by

$$t_n(\lambda) = \left(\frac{d-\lambda}{d}\right)^n .$$

Proof Suppose $q_n \in S_n$ and

$$\max_{\lambda\in F_a(d,0)} |q_n(\lambda)| < \max_{\lambda\in F_a(d,0)} |t_n(\lambda)| = \left(\frac{a}{|d|}\right)^n .$$

Then,

$$|q(\lambda)| < |t(\lambda)| = \frac{a}{|d|}^{n} ,$$

for every $\lambda \in F_a(d,0)$. By Rouché's theorem the polynomial $t_n(\lambda) - q_n(\lambda)$ has the same number of zeros inside $F_a(d,0)$ as the polynomial $t_n(\lambda)$ does. Notice that $t_n(0) - q_n(0) = 0$. Since $\lambda = 0$ is not in the interior of $F_a(d,0)$, $t_n(\lambda) - q_n(\lambda)$ is a polynomial of degree n with n+1 roots. We can conclude that $t_n(\lambda) = q_n(\lambda)$, and the theorem is proved.

If the region is bounded by an ellipse with real foci that does not contain the origin in its interior, we have the following result due to Clayton (Wrigley [25]).

Theorem 2.8 Let $0 < c \leq a \leq d$. The unique polynomial $t_n \in S_n$ such that

$$\max_{\lambda \in F_a(d,c)} |t_n(\lambda)| = \min_{s_n \in S_n} \max_{\lambda \in F_a(d,c)} |s_n(\lambda)|$$

is given by

$$t_n(\lambda) = P_n(\lambda) = \frac{T_n(\frac{d-\lambda}{c})}{T_n(\frac{d}{c})} ,$$

the associated scaled and translated Tchebychef polynomial.

Proof Omitted (Wrigley [25]).

This result cannot be extended to d and c with complex values (an easy example can be constructed), but it can be shown to be asymptotically true. For large n, $P_n(\lambda)$ tends very quickly to the optimal polynomial in S_n.

Lemma 2.9 Suppose $F_a(d,c)$ does not contain the origin in its interior. Let $t_n \in S_n$ be the unique polynomial such that

$$\max_{\lambda\in F_a(d,c)} |t_n(\lambda)| = \min_{s_n\in S_n} \max_{\lambda\in F_a(d,c)} |s_n(\lambda)|;$$

then,

$$\min_{\lambda\in F_a(d,c)} |P_n(\lambda)| \leq \max_{\lambda\in F_a(d,c)} |t_n(\lambda)| \leq \max_{\lambda\in F_a(d,c)} |P_n(\lambda)|.$$

Proof The second inequality is true by hypothesis. Suppose that

$$\max_{\lambda\in F_a(d,c)} |t_n(\lambda)| < \min_{\lambda\in F_a(d,c)} |P_n(\lambda)|;$$

then,

$$t_n(\lambda) < P_n(\lambda)$$

for every $\lambda \in F_a(d,c)$. By Rouché's theorem, the polynomial $P_n(\lambda) - t_n(\lambda)$ has the same number of zeros in the interior of $F_a(d,c)$ as $P_n(\lambda)$ does. $P_n(\lambda)$ has n zeros on the line segment joining the foci, d+c and d-c. Notice that $P_n(0) - t_n(0) = 0$. Since $\lambda = 0$ is not in the interior of $F_a(d,c)$, $P_n(\lambda) - t_n(\lambda)$ is a polynomial of degree n with n+1 zeros. We can conclude that $P_n(\lambda) = t_n(\lambda)$, and the lemma is proved.

Theorem 2.10 Suppose $F_a(d,c)$ does not include the origin in its interior. Let $t_n \in S_n$ be the unique polynomial such that

$$\max_{\lambda\in F_a(d,c)} (t_n(\lambda)) = \min_{s_n\in S_n} \max_{\lambda\in F_a(d,c)} (s_n(\lambda)).$$

Let

$$M(s_n) = \max_{\lambda \epsilon F_a(d,c)} |s_n(\lambda)| ;$$

then,

$$\lim_{n\to\infty} [M(t_n)^{\frac{1}{n}}] = \lim_{n\to\infty} [M(P_n)^{\frac{1}{n}}] .$$

Proof Let

$$m(P_n) = \min_{\lambda \epsilon F_a(d,c)} |P_n(\lambda)| .$$

From Lemma 2.9 we know that $m(P_n) \leq M(t_n) \leq M(P_n)$. It is sufficient to show that

$$\lim_{n\to\infty} [m(P_n)^{\frac{1}{n}}] = \lim_{n\to\infty} [M(P_n)^{\frac{1}{n}}] .$$

By Theorem 2.5 all points on the ellipse $F_a(d,c)$ have the same convergence factor; thus, we have

$$r(\lambda) = \lim_{n\to\infty} [m(P_n)^{\frac{1}{n}}] = \lim_{n\to\infty} [M(P_n)^{\frac{1}{n}}]$$

for every $\lambda \epsilon F_a(d,c)$. This proves the theorem.

Because of the nature of the cosh function, the asymptotic convergence factor is achieved very quickly; thus, the scaled and translated Tchebychef polynomials tend very quickly to the optimal polynomial in S_n.

As the focal length c approaches 0 the ellipse $F_a(d,c)$ is deformed into the circle $F_a(d,0)$. To show that the result for circles is compatible with the result for ellipses we have the following lemma:

<u>Lemma 2.11</u> If $P_n(\lambda) = \dfrac{T_n(\frac{d-\lambda}{c})}{T_n(\frac{d}{c})}$, then

$$\lim_{c \to 0} P_n(\lambda) = (\frac{d-\lambda}{d})^n .$$

<u>Proof</u> By the definition of cosh, we have

$$P_n(\lambda) = \frac{T_n(\frac{d-\lambda}{c})}{T_n(\frac{d}{c})} = \frac{e^{n \cosh^{-1}(\frac{d-\lambda}{c})} + e^{-n \cosh^{-1}(\frac{d-\lambda}{c})}}{e^{n \cosh^{-1}(\frac{d}{c})} + e^{-n \cosh^{-1}(\frac{d}{c})}} ,$$

which tends to

$$\frac{e^{n \cosh^{-1}(\frac{d-\lambda}{c})}}{e^{n \cosh^{-1}(\frac{d}{c})}}$$

as c decreases. By the log form of $\cosh^{-1}$ $(\cosh^{-1}(w) = \ln(w + (w^2-1)^{\frac{1}{2}}))$ this is

$$\left[\frac{(\frac{d-\lambda}{c}) + [(\frac{d-\lambda}{c})^2 - 1]^{1/2}}{d/c + [(d/c)^2 - 1]^{1/2}}\right]^n = \left[\frac{(d-\lambda) + [(d-\lambda)^2 - c^2]^{1/2}}{d + [d^2 - c^2]^{1/2}}\right]^n$$

which tends to $(\frac{d-\lambda}{d})^n$ as c decreases. This proves the lemma.

2.3 Convergence of $P_n^{(j)}(\lambda)$

Recall from Section 1.3 that if the matrix A has nonlinear elementary divisors, the derivatives of the sequence of polynomials must also converge to zero on the eigenvalues of A. In this section it will be shown that each derivative sequence of $P_n(\lambda)$ has the same region of convergence as the sequence $P_n(\lambda)$ does.

Theorem 2.12 Let $P_n(\lambda) = \dfrac{T_n(\frac{d-\lambda}{c})}{T_n(\frac{d}{c})}$. If λ is inside $F_0(d,c)$, the member of $\mathcal{F}(d,c)$ passing through the origin, then

$$\lim_{n\to\infty} P_n^{(j)}(\lambda) = 0 ,$$

for every j.

Proof Suppose λ is inside $F_0(d,c)$. Since the interior of $F_0(d,c)$ is an open set, there is some $\delta > 0$ such that $\Gamma = \{t/|t-\lambda| = \delta\}$ is inside $F_0(d,c)$. We have

$$P_n^{(j)}(\lambda) = \frac{j!}{2\pi i}\int_\Gamma \frac{P_n(t)}{(t-\lambda)^{j+1}}\,dt .$$

Let $|P_n(\lambda_n)| = \max_{\lambda\in\Gamma} |P_n(\lambda)|$; then,

$$|P_n^{(j)}(\lambda)| \leq \frac{j!}{2\pi}\int_\Gamma \frac{|P_n(t)|}{|t-\lambda|^{j+1}}\,dt$$

$$\leq \frac{j!}{2\pi}\,\frac{|P_n(\lambda_n)|}{(\delta)^{j+1}}\int_\Gamma dt$$

$$= j!\,\frac{|P_n(\lambda_n)|}{(\delta)^{j}} \,.$$

Since Γ is in the interior of $F_0(d,c)$ we know that

$$|P_n(\lambda_n)| \doteq r^n$$

for some $r < 1$, and the theorem is proved.

The speed of convergence of $|P_n(\lambda_n)|$ depends upon how close Γ is to the boundary, $F_0(d,c)$. There is a trade off between choosing smaller δ and the speed at which $|P_n(\lambda_n)|$ converges. One can, in fact, pick a new δ for each n and get

$$|P_n^{(j)}(\lambda)| \leq K(\lambda,j)\, n^j\, |P_n(\lambda)|$$

where $K(\lambda,j)$ is a constant depending on λ and j. From Section 2.1 we know that $|P_n(\lambda)| \doteq r(\lambda)^n$, so that

$$|P_n^{(j)}(\lambda)| \leq K(\lambda,j)\, n^j r(\lambda)^n \ .$$

We can conclude that when the ellipse $F_0(d,c)$ contains the spectrum of the matrix A in its interior, a gradient method based upon the associated scaled and translated Tchebychef polynomials will converge, although more slowly, in spite of the presence of nonlinear elementary divisors.

2.4 The Tchebychef Iteration

Because of the recursive property of the Tchebychef polynomials, the third criterion from Section 1.3 can be met. An iteration based on the scaled and translated Tchebychef polynomials, often called the Stiefel iteration (Stiefel [19]), can be carried out recursively, requiring the storage of only three vectors.

As before let x be the solution of the system $Ax = b$. Let x_n be the n^{th} iterative solution, let $e_n = x - x_n$ be the error at the n^{th} step, and let $r_n = b - Ax_n$ be the residual at the n^{th} step. If the iteration is to be based on the polynomials $P_n(\lambda) = \dfrac{T_n(\frac{d-\lambda}{c})}{T_n(\frac{d}{c})}$, then

$$e_n = P_n(A)\; e_0 \ .$$

We have

$$x_{n+1} - x_n = (P_n(A) - P_{n+1}(A))\, e_0 .$$

Let $Dx_n = x_{n+1} - x_n$. Since the Tchebychef polynomials satisfy:

$$T_0(z) = 1 ,$$

$$T_1(z) = z ,$$

$$T_{n+1}(z) = 2z\, T_n(z) - T_{n-1}(z),\; n > 0 ,$$

then for $n > 0$ we have

$$P_{n+1}(\lambda) = \frac{T_{n+1}(\frac{d-\lambda}{c})}{T_{n+1}(\frac{d}{c})} = \frac{2(\frac{d-\lambda}{c})\, T_n(\frac{d-\lambda}{c}) - T_{n-1}(\frac{d-\lambda}{c})}{T_n(\frac{d}{c})}$$

$$= -\frac{2}{c}\,\frac{\lambda T_n(\frac{d-\lambda}{c})}{T_{n+1}(\frac{d}{c})} + \frac{2\,\frac{d}{c}\, T_n(\frac{d-\lambda}{c})}{T_{n+1}(\frac{d}{c})} - \frac{T_{n-1}(\frac{d-\lambda}{c})}{T_{n+1}(\frac{d}{c})}$$

$$= -\left[\frac{2}{c}\,\frac{T_n(\frac{d}{c})}{T_{n+1}(\frac{d}{c})}\right] \lambda\, P_n(\lambda) + \left[\frac{2\,\frac{d}{c} T_n(\frac{d}{c})}{T_{n+1}(\frac{d}{c})}\right] P_n(\lambda) - \left[\frac{T_{n-1}(\frac{d}{c})}{T_{n+1}(\frac{d}{c})}\right] P_{n-1}(\lambda).$$

We have for $n > 0$

$$P_n(\lambda) - P_{n+1}(\lambda) = \left[\frac{2}{c}\,\frac{T_n(\frac{d}{c})}{T_{n+1}(\frac{d}{c})}\right] \lambda\, P_n(\lambda) + \left[1 - \frac{2\,\frac{d}{c} T_n(\frac{d}{c})}{T_{n+1}(\frac{d}{c})}\right] P_n(\lambda)$$

$$+ \left[\frac{T_{n-1}(\frac{d}{c})}{T_{n+1}(\frac{d}{c})}\right] P_{n-1}(\lambda)$$

$$= \left[\frac{2}{c}\,\frac{T_n(\frac{d}{c})}{T_{n+1}(\frac{d}{c})}\right] \lambda\, P_n(\lambda) + \left[\frac{T_{n-1}(\frac{d}{c})}{T_{n+1}(\frac{d}{c})}\right] (P_{n-1}(\lambda) - P_n(\lambda)).$$

We can write

$$Dx_n = (P_n(A) - P_{n+1}(A))\ e_0$$

$$= \frac{2}{c}\left[\frac{T_n(\frac{d}{c})}{T_{n+1}(\frac{d}{c})}\right] A \quad P_n(A)\ e_0 + \left[\frac{T_{n-1}(\frac{d}{c})}{T_{n+1}(\frac{d}{c})}\right] (P_{n-1}(A) - P_n(A))\ e_0 \ .$$

Notice that

$$r_n = A\ e_n = A P_n(A)\ e_0 \ ,$$

and that

$$Dx_{n-1} = (P_{n-1}(A) - P_n(A))\ e_0 \ .$$

We have then

$$Dx_n = \frac{2}{c}\left[\frac{T_n(\frac{d}{c})}{T_{n+1}(\frac{d}{c})}\right] r_n + \left[\frac{T_{n-1}(\frac{d}{c})}{T_{n+1}(\frac{d}{c})}\right] Dx_{n-1} \ ,$$

for $n < 0$. For $n = 0$ we have

$$Dx_0 = (P_0(A) - P_1(A))\ e_0 \ .$$

Since

$$P_0(\lambda) - P_1(\lambda) = 1 - (1 - \frac{\lambda}{d}) = \frac{\lambda}{d} \ ,$$

we have

$$Dx_0 = \frac{1}{d}\ Ae_0 = \frac{1}{d}\ r_0 \ .$$

The three term iteration becomes: given initial guess x_0 and parameters d and c, let

$$r_0 = b - A\,x_0 \,,$$

$$Dx_0 = \frac{1}{d}\, r_0 \,,$$

$$x_1 = x_0 + Dx_0 \,.$$

For $n > 0$, let

$$r_n = b - A\,x_n \,,$$

$$Dx_n = \frac{2}{c}\,\frac{T_n(\frac{d}{c})}{T_{n+1}(\frac{d}{c})}\, r_n + \frac{T_{n-1}(\frac{d}{c})}{T_{n+1}(\frac{d}{c})}\, Dx_{n-1} \,,$$

$$x_{n+1} = x_n + Dx_n \,.$$

The coefficients can be recursively generated and the iteration can be carried out with only x_i, r_i, Dx_i in storage.

If the spectrum of the matrix A lies in the right half plane, then it can be enclosed in an ellipse that does not contain the origin in its interior. The associated scaled and translated Tchebychef polynomials meet the criteria established in Section 1.3. They have minimal maximum modulous properties on ellipses, their derivative sequences also converge, and the iteration can be carried out by a three term recursion. The remainder of this thesis will be devoted to implementing an iteration based upon the Tchebychef polynomials.

3. CHOOSING OPTIMAL PARAMETERS

The spectrum of the matrix A can be enclosed in many different ellipses. In fact, given any family of ellipses, $\mathcal{F}(d,c)$, there is some member of the family that contains the spectrum of A in its interior. If the spectrum of A lies on the interior of $F_0(d,c)$, the member of the family $\mathcal{F}(d,c)$ passing through the origin, then the iteration based on the associated scaled and translated Tchebychef polynomials will converge. We would like to choose $\mathcal{F}(d,c)$ so that this convergence is optimal in some sense.

In Section 3.1 it will be shown that the choice of parameters which yield the best rate of convergence can be found as the solution of a mini-max problem. The mini-max problem will be restricted and restated in Section 3.2. In Section 3.3 it will be shown that if the matrix is real, the iteration can be carried out in real arithmetic.

3.1 The Mini-max Problem

Suppose d and c have been chosen. Let $F_s(d,c)$ be the smallest member of the family $\mathcal{F}(d,c)$ containing the spectrum of A in the closure of its interior. There is some eigenvalue, say λ_s, such that $\lambda_s \in F_s(d,c)$. From Section 2.1 we know that $F_s(d,c)$ is associated with a convergence factor, $r(\lambda_s)$, and that

$$r(\lambda_s) = \max_{\lambda_i} r(\lambda_i) \ ,$$

because all of the other eigenvalues are inside or on $F_s(d,c)$.

Recall that $r(\lambda)$ is also a function of d and c. We have

$$r(\lambda) = \left| e^{[\cosh^{-1}(\frac{d-\lambda}{c}) - \cosh^{-1}(\frac{d}{c})]} \right| .$$

If we use the log form of the $\cosh^{-1}$, then

$$r(\lambda) = \left| \frac{(\frac{d-\lambda}{c}) + ((\frac{d-\lambda}{c})^2 - 1)^{\frac{1}{2}}}{(\frac{d}{c}) + ((\frac{d}{c})^2 - 1)^{\frac{1}{2}}} \right|$$

$$= \left| \frac{(d-\lambda) + ((d-\lambda)^2 - c^2)^{\frac{1}{2}}}{d + (d^2 - c^2)^{\frac{1}{2}}} \right| .$$

One way to optimize the choice of d and c is to make $r(\lambda_s)$ as small as possible. The parameters d and c will then satisfy

$$\min_{d,c} \max_{\lambda_i} r(\lambda_i) = \min_{d,c} \max_{\lambda_i} \left| \frac{(d-\lambda_i) + ((d-\lambda_i)^2 - c^2)^{\frac{1}{2}}}{d + (d^2 - c^2)^{\frac{1}{2}}} \right| .$$

A more rigorous argument which yields this same mini-max problem is as follows. With a polynomial based gradient method the error is suppressed in accordance with the equation $e_n = P_n(A)\, e_0$. The following definition of rate of convergence is used by Young [26].

<u>Definition</u> The rate of convergence of a polynomial based gradient method applied to the system $Ax = b$ is

$$R(A) = -\log\,(\lim_{n\to\infty}(\|P_n(A)\|^{\frac{1}{n}})) .$$

We would like to choose d and c to make $R(A)$ as large as possible or, equivalently, to make $\lim_{n\to\infty} (\|P_n(A)\|^{\frac{1}{n}})$ as small as possible. Let

$$M(\lambda) = e^{[\cosh^{-1}(\frac{d-\lambda}{c}) - \cosh^{-1}(\frac{d}{c})]} .$$

If we use the log form of $\cosh^{-1}$ this becomes

$$M(\lambda) = \frac{(d-\lambda) + ((d-\lambda)^2 - c^2)^{\frac{1}{2}}}{d + (d^2 - c^2)^{\frac{1}{2}}} .$$

From Lemma 2.2 we have

$$P_n(\lambda) \doteq (M(\lambda))^n$$

for large n. Since $M(\lambda)$ is analytic in an open set containing the spectrum of A, there exists an operator $M(A)$. The eigenvalues of $M(A)$ are $M(\lambda_i)$ where λ_i is an eigenvalue of A (Dunford and Schwartz [4]). We have

$$P_n(A) \doteq (M(A))^n$$

for large n, so that

$$\lim_{n\to\infty} (\|P_n(A)\|^{\frac{1}{n}}) = \lim_{n\to\infty} (\|M(A)^n\|^{\frac{1}{n}})$$

$$= \text{spectral radius of } M(A) .$$

The spectral radius of $M(A)$ is

$$\max_{\lambda_i} |M(\lambda_i)| = \max_{\lambda_i} r(\lambda_i) .$$

The choice of d and c which yields the optimal rate of convergence is the solution to the mini-max problem above. Since $r(\lambda)$ is a function of d and c as well as λ, let us write the mini-max problem as

$$\min_{d,c} \max_{\lambda_i} r(\lambda_i, d, c) .$$

3.2 Restrictions

In this section we will show that if A is a real valued matrix, the mini-max problem can be restricted so that the maximum is taken over a subset of the eigenvalues and the minimum is taken over d and c such that d and c^2 are real.

In Section 3.1 we defined the ellipse $F_s(d,c)$ to be the smallest member of the family $\mathcal{F}(d,c)$ enclosing the spectrum of A. Since $F_s(d,c)$ is convex, the eigenvalue $\lambda_s \in F_s(d,c)$ has the property that it is on the boundary of the convex hull of the spectrum. In fact, it is a vertex of the smallest convex polygon enclosing the spectrum.

<u>Definition</u> Let $H = \{\lambda_i | \lambda_i$ is a vertex of the smallest convex polygon enclosing the spectrum of A$\}$. We will refer to H as the <u>hull</u> of the spectrum.

The elements of H completely determine the mini-max problem.

<u>Lemma 3.1</u> For any d, c

$$\max_{\lambda_i} r(\lambda_i, d, c) = \max_{\lambda_i \in H} r(\lambda_i, d, c) .$$

<u>Proof</u> Suppose there is a d and c and $\lambda_k \notin H$ such that

$$\max_{\lambda_i} r(\lambda_i, d, c) = r(\lambda_k, d, c) .$$

Since $\lambda_k \notin H$, we know that λ_k is not a vertex of the smallest convex polygon, P, enclosing the spectrum. Let $F_k(d,c)$ be the member of $\mathcal{F}(d,c)$ passing through λ_k. Since $F_k(d,c)$ is an ellipse and passes through a point that is either in the interior of P or on P between vertices, there is some vertex of P outside $F_k(d,c)$. From the definition of P, this vertex must be an eigenvalue, say λ_j. By Theorem 2.5

$$r(\lambda_j,d,c) > r(\lambda_k,d,c) \ .$$

This contradiction proves the lemma.

If A is a real valued matrix, then the eigenvalues of A are real or appear in complex conjugate pairs. The hull, H, of the spectrum is symmetric with respect to the real line. This motivates the following theorem, the proof of which is omitted.

<u>Theorem 3.2</u> If A is a real valued matrix and d and c satisfy the minimax problem,

$$\min_{d,c} \max_{\lambda_i \epsilon H} r(\lambda_i,d,c) \ ,$$

then the family $\mathcal{F}(d,c)$ is symmetric with respect to the real axis.

<u>Proof</u> Omitted.

If A is real we may restrict our search to those d and c which correspond to families of ellipses which are symmetric with respect to the real line. Such a family has foci that are either both real or are a complex conjugate pair. Since the foci are d+c and d-c, then d is real and c is either real or pure imaginary. In either case c^2 is real. Notice in the log form of the definition of $r(\lambda,d,c)$ in Section 3.1, that c appears only as c^2. Let $c2 = c^2$, and let

$$r(\lambda,d,c2) = \left| \frac{(d-\lambda) + ((d-\lambda)^2 - c2)^{\frac{1}{2}}}{d + (d^2 - c2)^{\frac{1}{2}}} \right| .$$

Since the families $\mathcal{F}(d,c)$ and $\mathcal{F}(d,-c)$ have the same foci, d+c and d-c, the parameters d and c2 uniquely determine the family $\mathcal{F}(d,c)$.

For A real we may restrict d and c2 to real values. In addition we may ignore those values of d and c2 for which convergence clearly does not occur.

Definition Let $R = \{(d,c2)/0 < d,\ c2 < d^2\}$.

Corollary 3.3 If A is a real valued matrix with eigenvalues in the right half plane the mini-max problem can be written

$$\min_{(d,c2)\in R}\ \max_{\lambda_i \in H}\ r(\lambda_i,d,c2) .$$

Proof It is clear from the discussion above that d and c2 may be restricted to the real numbers. By hypothesis, A has eigenvalues in the right half plane. If $d \leq 0$, then every eigenvalue of A would be outside $F_0(d,c)$, the ellipse passing through the origin. Convergence could not occur for this choice of d. If $d > 0$ and $c2 \geq d^2$, then $c \geq d$ and $d-c \leq 0 \leq d+c$. The family $\mathcal{F}(d,c)$ has one foci on each side of the origin. The ellipse $F_0(d,c)$ is the degenerate ellipse. Since $F_0(d,c)$ has no interior, there is no region of convergence.

Because A has its eigenvalues in the right half plane, there is some d and c2 for which convergence will occur. The solution of the mini-max problem is in the set R, and the lemma is proved.

Notice that for $(d,c2) \in R$ we have

$$r(\lambda,d,c2) = r(\bar{\lambda},d,c2) .$$

The maximum is completely determined by the eigenvalues with nonnegative imaginary part.

Definition Let $H^+ = \{\lambda_i \epsilon H/\mathrm{Im}(\lambda) \geq 0\}$. We will refer to H^+ as the positive hull of the spectrum.

Corollary 3.4 If A is a real valued matrix with eigenvalues in the right half plane, then the mini-max problem can be written

$$\min_{(d,c2)\epsilon R} \max_{\lambda_i \epsilon H^+} r(\lambda_i, d, c2) .$$

Proof The proof is clear from the discussion above.

A further reduction of the set H^+ is possible. We would like to find the smallest set of eigenvalues that completely determines $\max\limits_{\lambda_i} r(\lambda_i, d, c2)$ when $(d, c2) \epsilon R$.

Definition Let $K = \{\lambda_i \epsilon H^+$/there exists $(d, c2) \epsilon R$ such that $r(\lambda_i, d, c2) = \max\limits_{\lambda_i} r(\lambda_i, d, c2)\}$. The elements of K will be known as key elements.

Clearly, if $(d, c2) \epsilon R$, then

$$\max_{\lambda_i \epsilon K} r(\lambda_i, d, c2) = \max_{\lambda_i \epsilon H^+} r(\lambda_i, d, c2) .$$

Criteria to determine when an eigenvalue is in the set K are needed.

Lemma 3.5 If $\lambda_k \epsilon K$, then one of the following is true:

1. $\mathrm{Re}(\lambda_k) \leq \mathrm{Re}(\lambda_i)$ for every $\lambda_i \epsilon H^+$
2. $\mathrm{Re}(\lambda_k) \geq \mathrm{Re}(\lambda_i)$ for every $\lambda_i \epsilon H^+$
3. There exist λ_ℓ, $\lambda_m \epsilon H^+$ such that there is an ellipse, $F_k(d,c)$, with $(d, c2) \epsilon R$, passing through λ_k, λ_ℓ, and λ_m, containing the spectrum of A in the closure of its interior.

Proof Every point $(d, c2) \in R$ is associated with a family of ellipses, $\mathcal{F}(d,c)$. Let $F_k(d,c)$ be the member of $\mathcal{F}(d,c)$ passing through λ_k. As $(d,c2)$ is moved through the region R, $F_k(d,c)$ is continuously deformed.

If $\lambda_k \in K$ then there is some point $(d_1, c2_1) \in R$ such that

$$r(\lambda_k, d_1, c2_1) = \max_{\lambda_i} r(\lambda_i, d_1, c2_1) .$$

Since $r(\lambda_k, d_1, c2_1)$ is maximal we know from Theorem 2.5 that $F_k(d_1, c_1)$ contains the spectrum in the closure of its interior.

Suppose λ_k is the only eigenvalue on the ellipse $F_k(d_1, c_1)$. If λ_k does not satisfy 1. or 2. of the hypothesis, then there are at least two other eigenvalues in H^+, say λ_j and λ_n, such that

$$\mathrm{Re}(\lambda_j) < \mathrm{Re}(\lambda_k) < \mathrm{Re}(\lambda_n) .$$

Consider deforming the ellipse $F_k(d_1, c_1)$ into the degenerate ellipse, the line segment connecting λ_k and $\overline{\lambda}_k$, by moving $(d, c2)$ through R (see Figure 3.1). The eigenvalues λ_j and λ_n are inside $F_k(d_1, c_1)$ but outside the degenerate ellipse. One of the intermediate ellipses must have passed through another eigenvalue. As $(d, c2)$ moves from $(d_1, c2_1)$ let $(d_2, c2_2)$ be the first point such that the ellipse, $F_k(d_2, c_2)$ passes through another eigenvalue, say λ_ℓ. Since it was the first, $F_k(d_2, c_2)$ still encloses the spectrum, and, from Corollary 3.4, $\lambda_\ell \in H^+$.

Suppose λ_k and λ_ℓ are the only eigenvalues on the ellipse $F_k(d_2, c_2)$. We can move $(d, c2)$ through R in such a way that $F_k(d,c)$ passes through λ_k and λ_ℓ. As c2 gets negatively large the foci of the

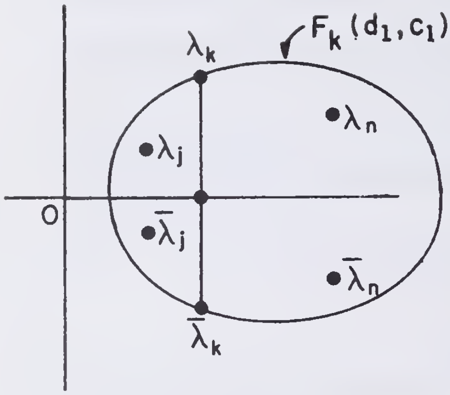

Figure 3.1

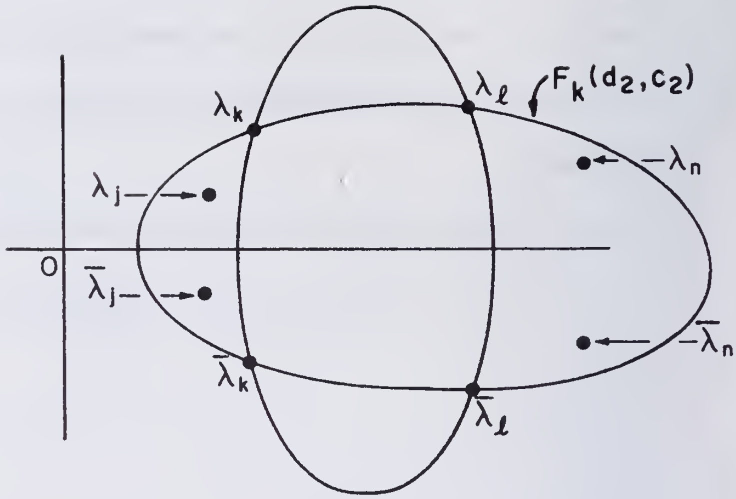

Figure 3.2

ellipse $F_k(d,c)$ have large imaginary part and the ellipse is deformed into the infinite column between $\mathrm{Re}(\lambda_k)$ and $\mathrm{Re}(\lambda_\ell)$ (see Figure 3.2). If $\mathrm{Re}(\lambda_k) < \mathrm{Re}(\lambda_\ell)$ then one of the intermediate ellipses must have passed through λ_j. If $\mathrm{Re}(\lambda_k) > \mathrm{Re}(\lambda_\ell)$, then one of the intermediate ellipses must have passed through λ_n. In either case, as $(d,c2)$ moves from $(d_2,c2_2)$ let $F_k(d_3,c_3)$ be the first ellipse to pass through a third eigenvalue, say λ_m. Since it is the first, $F_k(d_3,c_3)$ still encloses the spectrum, and, from Corollary 3.4, $\lambda_m \in H^+$. This proves the lemma.

The lemma provides criteria by which certain elements of the set H^+ can be ignored. The implementation of these criteria will arise naturally from the algorithm to be presented at the end of Chapter 4.

The results of this section are summed up in the following theorem.

Theorem 3.6 If A is a real valued matrix with eigenvalues in the right half plane, then the parameters d and c which yield the optimal rate of convergence can be found in terms of d and c2 = c^2 as the solution of the mini-max problem

$$\min_{(d,c2)\in R} \; \max_{\lambda_i \in K} \; r(\lambda_i, d, c2) \; .$$

3.3 Real Arithmetic

In this section it will be shown that if d and c2 are real, the iteration based upon the associated scaled and translated Tchebychef polynomials can be performed in real arithmetic, even when the scaling parameter c is pure imaginary.

Theorem 3.7 If d and c2 are real, then

$$P_n(\lambda) = \frac{T_n\left(\frac{d-\lambda}{c}\right)}{T_n\left(\frac{d}{c}\right)}$$

is a polynomial in λ with real coefficients.

Proof Since c2 = c^2, if c2 $\geq$ 0, then d and c are both real, and $P_n(\lambda)$ has real coefficients. If c2 < 0, then c = is where s is real. We have

$$P_n(\lambda) = \frac{T_n\left(\frac{d-\lambda}{is}\right)}{T_n\left(\frac{d}{is}\right)} \; .$$

If n is even, then $T_n(z)$ is an even polynomial. Each term raises $\left(\frac{d-\lambda}{is}\right)$ to an even power, so $T_n\left(\frac{d-\lambda}{is}\right)$ has real coefficients and $T_n\left(\frac{d}{is}\right)$ is real.

If n is odd, then $T_n(z)$ is an odd polynomial. Each term raises $(\frac{d-\lambda}{is})$ to an odd power, so i can be factored out of each term. Likewise, $T_n(\frac{d}{is})$ is pure imaginary. The quotient, $P_n(\lambda)$, has real coefficients, and the theorem is proved.

From Section 2.5, the iteration based upon the scaled and translated Tchebychef polynomials is as follows: to solve $Ax = b$, given x_0, d, c, let

$$r_0 = b - Ax_0$$

$$Dx_0 = \frac{1}{d} r_0$$

$$x_1 = x_0 + Dx_0 \ .$$

For $n \geq 1$, let

$$r_n = b - Ax_n$$

$$Dx_n = \left[\frac{2}{c} \frac{T_n(\frac{d}{c})}{T_{n+1}(\frac{d}{c})}\right] r_n + \left[\frac{T_{n-1}(\frac{d}{c})}{T_{n+1}(\frac{d}{c})}\right] Dx_{n-1}$$

$$x_{n+1} = x_n + Dx_n \ .$$

The iteration parameters are given in terms of d, c and $T_n(\frac{d}{c})$. If c is pure imaginary the computation seems to require complex arithmetic. The parameters can be generated recursively, however, in terms of d and c2.

Theorem 3.8 If d and c2 are real then the iteration based upon the associated scaled and translated Tchebychef polynomials can be performed in real arithmetic.

Proof For $n \geq 1$ let

$$P1(n) = \frac{2}{c} \frac{T_n(\frac{d}{c})}{T_{n+1}(\frac{d}{c})} \ ,$$

$$P2(n) = \frac{T_{n-1}(\frac{d}{c})}{T_{n+1}(\frac{d}{c})} .$$

For $n = 1$ we have

$$P1(1) = \frac{T_1(\frac{d}{c})}{T_2(\frac{d}{c})} = \frac{2}{c}\,\frac{\frac{d}{c}}{2(\frac{d}{c})^2-1} = \frac{2d}{2d^2 - c2} ,$$

$$P2(1) = \frac{T_0(\frac{d}{c})}{T_2(\frac{d}{c})} = \frac{1}{2(\frac{d}{c})^2-1} = \frac{c2}{2d^2 - c2} .$$

If we use the recursive formulas for the Tchebychef polynomials (see Section 2.1), we have for $n > 1$

$$P1(n) = \frac{2}{c}\,\frac{T_n(\frac{d}{c})}{T_{n+1}(\frac{d}{c})} = \frac{1}{d}\,\frac{2\frac{d}{c}\,T_n(\frac{d}{c})}{T_{n+1}(\frac{d}{c})}$$

$$= \frac{1}{d}\left(\frac{T_{n+1}(\frac{d}{c}) + T_{n-1}(\frac{d}{c})}{T_{n+1}(\frac{d}{c})}\right) = \frac{1}{d}(1 + P2(n)) ,$$

$$P2(n) = \frac{T_{n-1}(\frac{d}{c})}{T_{n+1}(\frac{d}{c})} = \frac{T_{n-1}(\frac{d}{c})}{2\frac{d}{c}\,T_n(\frac{d}{c}) - T_{n-1}(\frac{d}{c})}$$

$$= \frac{\dfrac{T_{n-1}(\frac{d}{c})}{T_n(\frac{d}{c})}}{2\dfrac{d}{c} - \dfrac{T_{n-1}(\frac{d}{c})}{T_n(\frac{d}{c})}} = \frac{\dfrac{2}{c}\,\dfrac{T_{n-1}(\frac{d}{c})}{T_n(\frac{d}{c})}}{4\dfrac{d}{c^2} - \dfrac{2}{c}\,\dfrac{T_{n-1}(\frac{d}{c})}{T_n(\frac{d}{c})}}$$

$$= \frac{P1(n-1)}{4\frac{d}{c^2} - P1(n-1)} = \frac{c2\ P1(n-1)}{4d - c2\ P1(n-1)} .$$

If d and c2 are real, then P1(1) and P2(1) are real. By induction

$$P2(n) = \frac{c2\ P1(n-1)}{4d - c2\ P1(n-1)},$$

$$P1(n) = \frac{1}{d}(1 + P2(n))$$

are real for $n > 1$. The iteration can be performed in real arithmetic, which proves the theorem.

4. SOLVING THE MINI-MAX PROBLEM

If the eigenvalues in the positive hull, H^+, are known, then, as was shown in Chapter 3, the optimal parameters can be found as the point that minimizes the maximum of a finite number of real valued functions of two real variables. Consider each function, $r(\lambda_i, d, c2)$, to be a surface above the d,c2-plane. Section 4.1 will show that the mini-max solution is either a local minimum of one of the surfaces, a local minimum along the intersection of two surfaces, or a point where three surfaces intersect. Section 4.2 will show that each surface has only one local minimum in R. It will be found explicitly. The local minimum along the intersection of two surfaces will be found in Section 4.3 as the root of a fifth degree polynomial. In Section 4.4 the unique intersection of three surfaces will be found, when it exists, and existence criteria will be established. An algorithm to find the mini-max solution among the possible candidates will be developed in Section 4.5. It will be assumed that $\{\lambda_i\}$ are eigenvalues of A and that $Re(\lambda_i) > 0$, $I_m(\lambda_i) \geq 0$.

4.1 The Alternative Theorem

The following theorem will enable us to solve the mini-max problem by looking at three or fewer functions at a time.

<u>Theorem 4.1</u> (Alternative Theorem) If $\{f_i(x,y)\}$ is a finite set of real valued functions of two real variables, each of which is continuous on a closed and bounded region S and

$$M(x,y) = \max_i f_i(x,y) ,$$

then $M(x,y)$ takes on a minimum at some point (x_0,y_0) in the region S. If (x_0,y_0) is in the interior of S, then one of the following holds:

1. The point (x_0,y_0) is a local minimum of $f_i(x,y)$ for some i such that $M(x_0,y_0) = f_i(x_0,y_0)$.

2. The point (x_0,y_0) is a local minimum among the locus $\{(x,y)\epsilon S/f_i(x,y) = f_j(x,y)\}$ for some i and j such that $M(x_0,y_0) = f_i(x_0,y_0) = f_j(x_0,y_0)$.

3. The point (x_0,y_0) is such that for some i, j, and k, $M(x_0,y_0) = f_i(x_0,y_0) = f_j(x_0,y_0) = f_k(x_0,y_0)$.

Proof Since $M(x,y)$ is continuous on S, it takes on its minimum at some point (x_0,y_0) in S. In the following all neighborhoods of (x_0,y_0) will be understood to be in the interior of S.

If there is a neighborhood of (x_0,y_0) such that $M(x,y) = f_i(x,y)$ in the neighborhood, then (x_0,y_0) is a local minimum of $f_i(x,y)$ and $M(x_0,y_0) = f_i(x_0,y_0)$.

Suppose that there is no neighborhood of (x_0,y_0) in which the maximum, $M(x,y)$, is determined by only one function but there is a neighborhood in which $M(x,y)$ is determined by two functions, say $f_i(x,y)$ and $f_j(x,y)$. Since $f_i(x,y)$ and $f_j(x,y)$ are continuous, then $M(x_0,y_0) = f_i(x_0,y_0) = f_j(x_0,y_0)$. Clearly, (x_0,y_0) is a local minimum in the locus $\{(x,y)\epsilon S/f_i(x,y) = f_j(x,y)\}$.

Suppose that in every neighborhood of (x_0,y_0) the maximum, $M(x,y)$, is determined by at least three functions. Since they are all continuous, there are three functions, say $f_i(x,y)$, $f_j(x,y)$, and $f_k(x,y)$,

such that $M(x_0,y_0) = f_i(x_0,y_0) = f_j(x_0,y_0) = f_k(x_0,y_0)$. This proves the theorem.

Since the spectrum of A lies in the right half plane, it can be enclosed in an ellipse symmetric with respect to the real line that does not include the origin. Equivalently, there is some point in the region R such that

$$\max_{\lambda_i} r(\lambda_i,d,c2) < 1 .$$

In the next section it will be shown that for each λ_i, $r(\lambda_i,d,c2) \geq 1$ on the boundary of R. There is, therefore, a closed and bounded subregion $S \subset R$ which contains the mini-max solution in its interior, and we can apply the Alternative Theorem to the mini-max problem.

4.2 Minimum Point of a Single Function

In this section the one local minimum of the function $r(\lambda_i,d,c2)$ in the region R will be found in terms of λ_i.

Each point $(d,c2) \in R$ is associated with a family of ellipses in the λ-plane whose members are the level lines of $r(\lambda,d,c2)$. Let $F_i(d,c)$ be the ellipse in this family passing through λ_i. From Theorem 2.5 we know that $r(\lambda,d,c2)$ takes on the same value at each $\lambda \in F_i(d,c)$. In particular, consider λ_0 in Figure 4.1. Since $(d-\lambda_0)^2 \geq c2$ we have

$$r(\lambda_i,d,c2) = r(\lambda_0,d,c2) = \frac{(d-\lambda_0) + ((d-\lambda_0)^2 - c2)^{\frac{1}{2}}}{d + (d^2 - c2)^{\frac{1}{2}}} .$$

If we let $(d-\lambda_0)^2 = a2$, then every point $\lambda = x + iy$ on the ellipse

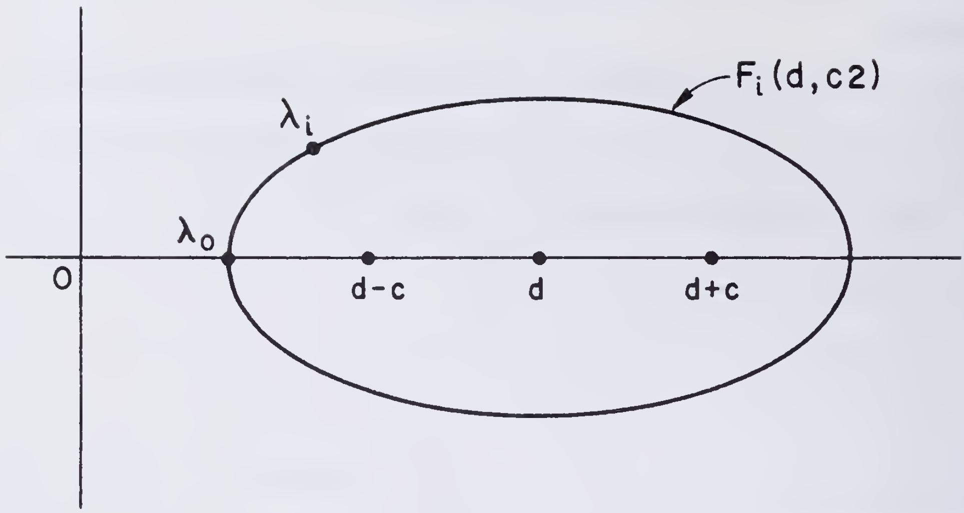

Figure 4.1

$F_i(d,c)$ will satisfy

$$\frac{(d-x)^2}{a2} + \frac{y^2}{a2-c2} = 1 \ .$$

Letting $\lambda_i = x_i + iy_i$, we can write

$$r(\lambda_i, d, c2) = \frac{(a2)^{\frac{1}{2}} + (a2-c2)^{\frac{1}{2}}}{d + (d^2-c2)^{\frac{1}{2}}} \ ,$$

subject to the constraint

$$\frac{(d-x_i)^2}{a2} + \frac{y_i^2}{a2-c2} = 1 \ .$$

(If the ellipse $F_i(d,c)$ is degenerate, the above constraint does not hold. If $y_i \neq 0$, the degenerate case gives $a2 = 0$. If $y_i = 0$, the degenerate case gives $a2 = c2$. In any case, $a2$ varies continuously with d and $c2$.)

It is clear that if λ_i is in the right half plane, we can pick $(d,c2) \in R$ such that the ellipse $F_i(d,c)$ does not contain the origin. Equivalently, there is some $(d,c2) \in R$ such that $r(\lambda_i,d,c2) < 1$. The following lemma will show that we need only look in the interior of R for the local minima of $r(\lambda_i,d,c2)$.

<u>Lemma 4.2</u> If $\mathrm{Re}(\lambda_i) > 0$, then $r(\lambda_i,d,c2) \geq 1$ on the boundary of R.

<u>Proof</u> We have $R = \{(d,c2)/0 < d,\ c2 < d^2\}$. From Corollary 3.3 we know that $r(\lambda_i,d,c2) \geq 1$ for $d = 0$ and $c2 = d^2$. From the constraint equation we have

$$(d-x_i)^2 \leq a2 \,,$$
$$y_i^2 \leq a2 - c2 \,.$$

If $c2 \leq (d-x_i)^2$, we can write

$$\frac{|d-x_i| + |((d-x_i)^2 - c2)^{\frac{1}{2}}|}{d + (d^2-c2)^{\frac{1}{2}}} \leq r(\lambda_i,d,c2) \,.$$

If $(d-x_i)^2 < c2 < d^2$, we can write

$$\frac{|d-x_i| + y_i}{d + (d^2-(d-x_i)2)^{\frac{1}{2}}} \leq r(\lambda_i,d,c2) \,.$$

In either case we have

$$\lim_{\substack{(d,c2)\to\infty \\ (d,c2)\in R}} r(\lambda_i,d,c2) \geq 1 \,,$$

which proves the lemma.

Since $r(\lambda_i, d, c2) \geq 1$ on the boundary of R, which would cause the iteration to diverge, we are only interested in local minima that occur in the interior of R. The next two theorems give the minimum of $r(\lambda_i, d, c2)$ in terms of λ_i. Theorem 4.3 treats the case $\lambda_i = x_i + iy_i$, $y_i \neq 0$. Theorem 4.4 treats the case $\lambda_i = x_i$.

Theorem 4.3 If $\lambda_i = x_i + iy_i$, $y_i \neq 0$, there is only one local minimum of the function $r(\lambda_i, d, c2)$ in the region R. It occurs at the point $(d, c2) = (x_i, -y_i^2)$, and at this point

$$r(\lambda_i, x_i, -y_i^2) = \frac{y_i}{x_i + (x_i^2 + y_i^2)^{\frac{1}{2}}} .$$

Proof In the form

$$r(\lambda_i, d, c2) = \frac{(a2)^{\frac{1}{2}} + (a2-c2)^{\frac{1}{2}}}{d + (d^2-c2)^{\frac{1}{2}}} ,$$

it is clear that $r(\lambda_i, d, c2)$ is a continuous function of d, c2, and a2 for values of d, c2 in R and continuously differentiable except where

$$a2 = 0,$$

$$a2 - c2 = 0 .$$

From the constraint equation,

$$\frac{(d-x_i)^2}{a2} + \frac{y_i^2}{a2-c2} = 1 ,$$

it is clear that a2 is a continuously differentiable function of d and c2 except where

$$a2 = 0,$$

$$a2 - c2 = 0 .$$

Since $y_i \neq 0$ and $a2 - c2 \geq y_i^2$, we have $a2 - c2 > 0$ in R. Now $a2 = 0$ when

$$d = x_i \quad \text{and} \quad c2 < -y_i^2 \ .$$

This is the ray shown in Figure 4.2 and corresponds to the degenerate ellipse through λ_i.

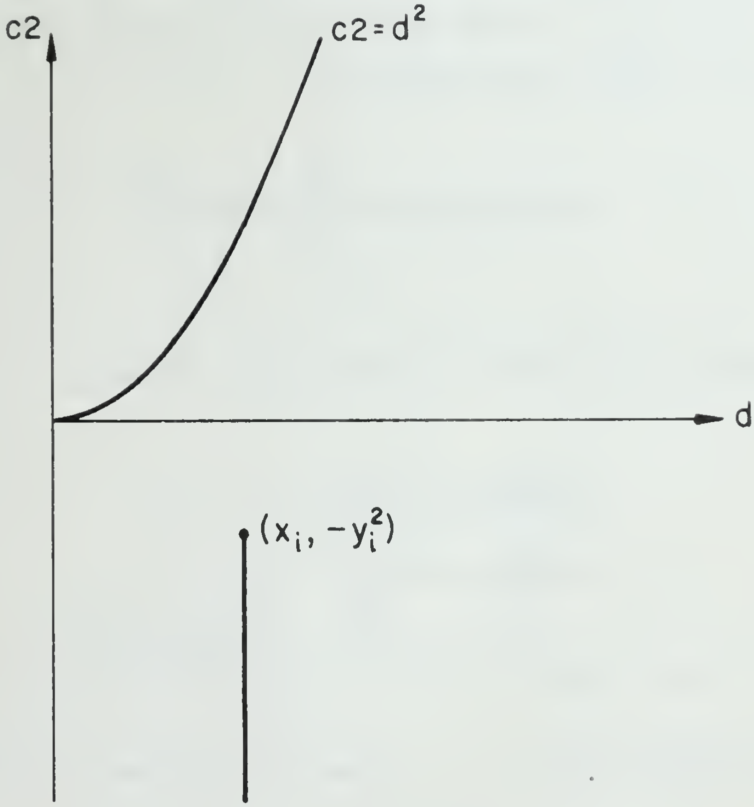

Figure 4.2

Any local minimum of $r(\lambda_i, d, c2)$ must occur at a critical point. A critical point is a point at which either all directional derivatives are zero, or some directional derivative does not exist. We have shown that the points on the ray above are critical points. It will be shown that these are the only critical points so that any local minimum must lie

on the ray. It will then be shown that the only local minimum on this ray is at the point $(x_i, -y_i^2)$.

First, we divide R into two regions: $d > x_i$ and $d < x_i$.

Case I: $d > x_i$.

Let $c2 = \rho d^2$. If $\rho < 1$ this describes a curve through R. Every point in R is on some curve $c2 = \rho d^2$, $\rho < 1$. It will be shown that there is only one possible critical point on this curve.

a) If $\rho > 0$, then $c2 > 0$ along the curve $c2 = \rho d^2$. Let $\beta = \frac{a2}{c2}$, $\gamma = \frac{1}{\rho} = \frac{d^2}{c2}$. We can write

$$r(\lambda_i, d, c2) = \frac{(\beta)^{\frac{1}{2}} + (\beta-1)^{\frac{1}{2}}}{(\gamma)^{\frac{1}{2}} + (\gamma-1)^{\frac{1}{2}}} .$$

The directional derivative along this curve becomes

$$r' = \frac{\frac{1}{2}\frac{\beta'}{(\beta)^{\frac{1}{2}}} + \frac{1}{2}\frac{\beta'}{(\beta-1)^{\frac{1}{2}}}}{(\gamma)^{\frac{1}{2}} + (\gamma-1)^{\frac{1}{2}}} = \frac{r}{2}\,\frac{\beta'}{(\beta)^{\frac{1}{2}}(\beta-1)^{\frac{1}{2}}} .$$

For $r' = 0$ we must have $\beta' = 0$. From the constraint equation we have

$$\frac{2(d-x_i)}{\beta} - \frac{(d-x_i)^2}{(\beta)^2}\beta' - \frac{(y_i)^2}{(\beta-1)^2}\beta' = 2\rho d ,$$

so that $\beta' = 0$ gives

$$\beta = \frac{d-x_i}{\rho d} ,$$

or

$$a2 = d(d-x_i) .$$

Plugging this back into the constraint equation we get

$$d = \left(\frac{1}{1-\rho}\right)\left(\frac{x_i^2 + y_i^2}{x_i}\right) ,$$

$$c2 = \rho d^2 ,$$

$$a2 = d(d-x_i) .$$

This is the only possible critical point along the curve $c2 = \rho d^2$, $0 < \rho < 1$.

b) If $\rho < 0$, then $c2 < 0$ along the curve $c2 = \rho d^2$. We can write

$$r(\lambda_i, d, c2) = \frac{(-\beta)^{\frac{1}{2}} + (1-\beta)^{\frac{1}{2}}}{(-\gamma)^{\frac{1}{2}} + (1-\gamma)^{\frac{1}{2}}} .$$

Taking the directional derivative along this curve we get

$$r' = \frac{r}{2} \frac{-\beta'}{(-\beta)^{\frac{1}{2}}(1-\beta)^{\frac{1}{2}}} .$$

Again we have that $r' = 0$ implies $\beta' = 0$, and again the only point on the curve satisfying this is

$$d = \left(\frac{1}{1-\rho}\right)\left(\frac{x_i^2 + y_i^2}{x_i}\right) ,$$

$$c2 = \rho d^2 ,$$

$$a2 = d(d-x_i) .$$

c) If $\rho = 0$, then $c2 = 0$ along the curve $c2 = \rho d^2$. We can write

$$r(\lambda_i, d, c2) = \frac{(a2)^{\frac{1}{2}}}{d} .$$

Taking the directional derivative along the curve we get

$$r' = \frac{1}{2} \frac{a2'}{(a2)^{\frac{1}{2}}} \cdot \frac{1}{d} - \frac{(a2)^{\frac{1}{2}}}{d^2} .$$

The constraint equation for $c2 = 0$ becomes

$$(d-x_i)^2 + y_i^2 = a2 ,$$

so

$$a2' = 2(d-x_i) .$$

Substituting this in the equation above we get

$$r' = \frac{1}{(a2)^{\frac{1}{2}} d^2} (d(d-x_i) - a2) .$$

If $r' = 0$, then $a2 = d(d-x_i)$. Using the constraint equation for $c2 = 0$, we have

$$d(d-x_i) = (d-x_i)^2 + y_i^2 ,$$

or

$$d = \frac{x_i^2 + y_i^2}{x_i} \ ,$$

$$c2 = 0 \ ,$$

$$a2 = d(d-x_i) \ ,$$

as the only possible critical point along the curve $c2 = 0$.

On each curve $c2 = \rho d^2$, $\rho < 1$, there is only one possible critical point. The locus of all of these possible critical points is the curve

$$c2 = d(d-\eta) \ ,$$

where

$$\eta = \frac{x_i^2 + y_i^2}{x_i} \ ,$$

and on this curve

$$a2 = d(d-x_i) \ .$$

Figure 4.3 shows the curve $c2 = d(d-\eta)$. The possible critical point on any curve $c2 = \rho d^2$ is where the two curves intersect.

Along the curve $c2 = d(d-\eta)$, $d > x_i$ we can write

$$r(\lambda_i, d, c2) = \frac{(d(d-x_i))^{\frac{1}{2}} + (d(d-x_i) - d(d-\eta))^{\frac{1}{2}}}{d + (d^2 - d(d-\eta))^{\frac{1}{2}}}$$

$$= \frac{(d-x_i)^{\frac{1}{2}} + (\eta-x_i)^{\frac{1}{2}}}{d^{\frac{1}{2}} + (\eta)^{\frac{1}{2}}} \ .$$

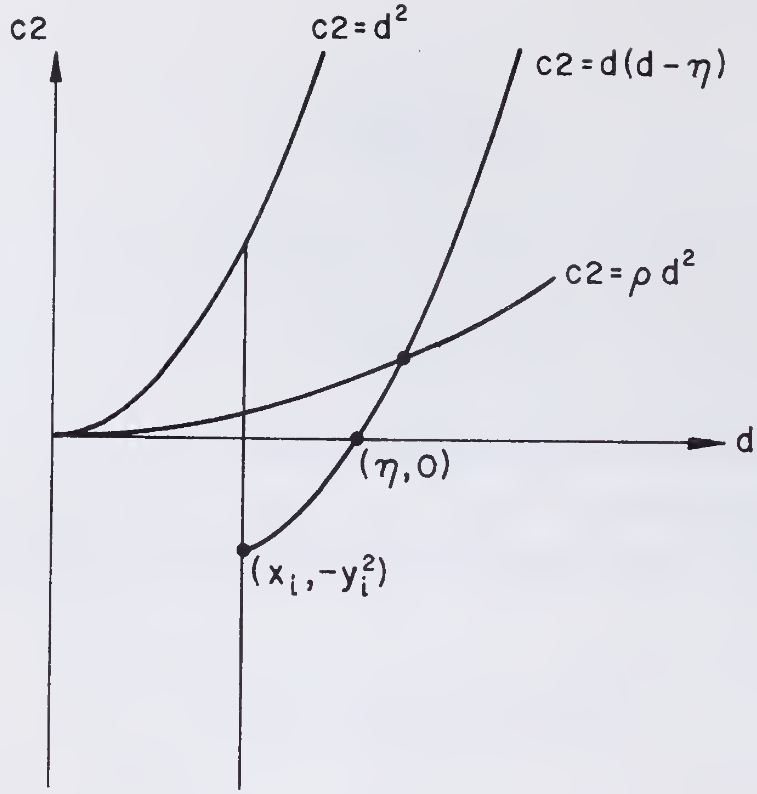

Figure 4.3

The directional derivative along the curve is

$$r' = \frac{\frac{1}{2}\,\frac{1}{(d-x_i)^{\frac{1}{2}}}\,(d^{\frac{1}{2}}+\eta^{\frac{1}{2}}) - \frac{1}{2}\,\frac{1}{d^{\frac{1}{2}}}\left((d-x_i)^{\frac{1}{2}} + (\eta-x_i)^{\frac{1}{2}}\right)}{(d^{\frac{1}{2}} + \eta^{\frac{1}{2}})^2}$$

$$= \frac{1}{2}\,\frac{1}{(d^{\frac{1}{2}}+\eta^{\frac{1}{2}})^2 (d-x_i)^{\frac{1}{2}}\, d^{\frac{1}{2}}} \cdot \left(d^{\frac{1}{2}}(d^{\frac{1}{2}}+\eta^{\frac{1}{2}}) - (d-x_i)^{\frac{1}{2}}\left((d-x_i)^{\frac{1}{2}} + (\eta-x_i)^{\frac{1}{2}}\right)\right) .$$

Since $d > d-x_i$ and $\eta > \eta-x_i$, then

$$d^{\frac{1}{2}}(d^{\frac{1}{2}}+\eta^{\frac{1}{2}}) > (d-x_i)^{\frac{1}{2}}((d-x_i)^{\frac{1}{2}} + (\eta-x_i)^{\frac{1}{2}}) .$$

We can conclude that $r' > 0$ for $d > x_i$ and that there are no critical points in the subregion $d > x_i$.

Case II: $d < x_i$.

As in Case I we look along the curves $c2 = \rho d^2$. Taking directional derivatives along the curves we again find that $r' = 0$ implies $a2 = d(d-x_i)$. Since $a2 > 0$ for $d \neq x_i$ and $d < x_i$ in this subregion, this condition is never met. We can conclude that there are no critical points in the region $d < x_i$.

We have now isolated the possible local minima to the line $d = x_i$. From the constraint equation we see that if $d = x_i$, then

$$a2 = \begin{cases} y_i^2 + c2 & \text{for} \quad x_i^2 > c2 > -y_i^2 , \\ 0 & \text{for} \quad -y_i^2 \geq c2 . \end{cases}$$

Case I: $x_i^2 > c2 > -y_i^2$, $d = x_i$.

We can write

$$r(\lambda_i, d, c2) = \frac{(y_i^2 + c2)^{\frac{1}{2}} + y_i}{x_i + (x_i^2 - c2)^{\frac{1}{2}}} .$$

Taking the derivative with respect to c2 we get

$$r' = \frac{\frac{1}{2}\dfrac{1}{(y_i^2+c2)^{\frac{1}{2}}}(x_i+(x_i^2-c2)^{\frac{1}{2}}) + \frac{1}{2}\dfrac{1}{(x_i^2-c2)^{\frac{1}{2}}}((y_i^2+c2)^{\frac{1}{2}}+y_i)}{(x_i + (x_i^2 - c2)^{\frac{1}{2}})^2} .$$

Since each term is positive, then $r' > 0$, and $r(\lambda_i, d, c2)$ is increasing as c2 increases from $c2 = -y_i^2$ along the line $d = x_i$.

Case II: $c2 < -y_i^2$, $d = x_i$.

We can write

$$r(\lambda_i, d, c2) = \frac{(-c2)^{\frac{1}{2}}}{x_i + (x_i - c2)^{\frac{1}{2}}} .$$

Taking the derivative with respect to c2 we have

$$r' = \frac{\frac{1}{2}\frac{-1}{(-c2)^{\frac{1}{2}}}(x_i + (x_i^2 - c2)^{\frac{1}{2}}) - \frac{1}{2}\frac{-1}{(x_i^2 - c2)^{\frac{1}{2}}}(-c2)^{\frac{1}{2}}}{(x_i + (x_i^2 - c2)^{\frac{1}{2}})^2}$$

$$= -\frac{(x_i^2 - c2)^{\frac{1}{2}}(x_i + (x_i^2 - c2)^{\frac{1}{2}}) + c2}{2(x_i + (x_i^2 - c2)^{\frac{1}{2}})^2 (x_i^2 - c2)^{\frac{1}{2}}(-c2)^{\frac{1}{2}}}$$

$$= -\frac{x_i(x_i^2 - c2)^{\frac{1}{2}} + x_i^2}{2(x_i + (x_i^2 - c2)^{\frac{1}{2}})^2 (x_i - c2)^{\frac{1}{2}}(-c2)^{\frac{1}{2}}}$$

$$= -\frac{x_i}{2(x_i + (x_i^2 - c2)^{\frac{1}{2}})(x_i - c2)^{\frac{1}{2}}(-c2)^{\frac{1}{2}}} .$$

Since each term is positive, then $r' < 0$, and $r(\lambda_i, d, c2)$ is decreasing as $c2$ increases toward $c2 = -y_i^2$ along the line $d = x_i$.

We can conclude that the only local minimum of $r(\lambda_i, d, c2)$ occurs at the point $(d, c2) = (x_i, -y_i^2)$ and at this point

$$r(\lambda_i, x_i, -y_i^2) = \frac{y_i}{x_i + (x_i^2 + y_i^2)^{\frac{1}{2}}} .$$

This proves the theorem.

<u>Theorem 4.4</u> If $\lambda_i = x_i$, then there is only one local minimum of the function $r(\lambda_i, d, c2)$ in the region R. It is at the point $(d, c2) = (x_i, 0)$, and at this point

$$r(\lambda_i, x_i, 0) = 0 .$$

<u>Proof</u> Since $\lambda_i = x_i$, we have $a2 = (d-x_i)^2$ except for the degenerate case. The degenerate case occurs when $c2 \geq (d-x_i)^2$, so that $\lambda_i = x_i$ is between the foci, $d+c$ and $d-c$. If $c2 \geq (d-x_i)^2$, then $a2 = c2$. We can write

$$r(\lambda_i, d, c2) = \begin{cases} \dfrac{|d-x_i| + ((d-x_i)^2 - c2)^{\frac{1}{2}}}{d + (d^2 - c2)^{\frac{1}{2}}} & \text{for } (d-x_i)^2 > c2 , \\ \\ \dfrac{(c2)^{\frac{1}{2}}}{d + (d^2 - c2)^{\frac{1}{2}}} & \text{for } (d-x_i)^2 \leq c2 . \end{cases}$$

In this form it is clear that $r(\lambda_i, d, c2)$ is continuous in R and continuously differentiable except where

$$c2 = (d-x_i)^2 ,$$

and where

$$d = x_i \quad \text{for} \quad c2 < 0 .$$

As in the proof of Theorem 4.3, it will be shown that the only critical points of the function $r(\lambda_i, d, c2)$ in R are those described above. Divide R into subregions as shown in Figure 4.4. Region I is where $c2 > (d-x_i)^2$, which corresponds to the degenerate ellipse. In this region we have $a2 = c2$. Region II is where $c2 < (d-x_i)^2$, $d > x_i$, and Region III is where $c2 < (d-x_i)^2$, $d < x_i$. In these regions we have $a2 = (d-x_i)^2$.

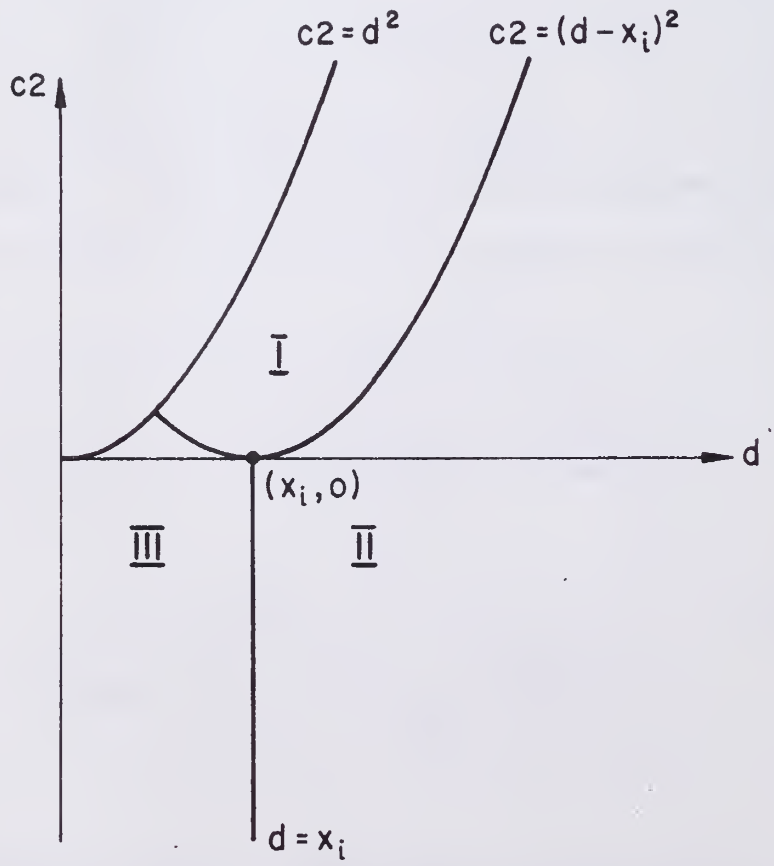

Figure 4.4

Case I: $c2 > (d-x_i)^2$.

In this subregion we have

$$r(\lambda_i, d, c2) = \frac{(c2)^{\frac{1}{2}}}{d + (d^2 - c2)^{\frac{1}{2}}} .$$

Taking the partial derivative with respect to c2, we get

$$r' = \frac{\frac{1}{2} \frac{1}{(c2)^{\frac{1}{2}}} (d + (d^2 - c2)^{\frac{1}{2}}) + \frac{1}{2} \frac{1}{(d^2 - c2)^{\frac{1}{2}}} (c2)^{\frac{1}{2}}}{(d + (d^2 - c2)^{\frac{1}{2}})^2}$$

$$= \frac{1}{2} \frac{(d + (d^2 - c2)^{\frac{1}{2}})(d^2 - c2)^{\frac{1}{2}} + c2}{(d + (d^2 - c2)^{\frac{1}{2}})^2 (c2)^{\frac{1}{2}} (d^2 - c2)^{\frac{1}{2}}} .$$

Since each of the terms is positive, we have $r' > 0$ in this region which implies that there are no critical points in this region.

Case II: $(d-x_i)^2 > c2$, $d > x_i$.

Case III: $(d-x_i)^2 > c2$, $d < x_i$.

These two cases will be treated together. In these regions we have $a2 = (d-x_i)^2$. As in the proof of Theorem 4.3, consider the curves $c2 = \rho d^2$, $\rho < 1$. Let

$$\beta = \frac{a2}{c2} = \frac{(d-x_i)^2}{\rho d^2} .$$

By the same argument used in Theorem 4.3, the directional derivative along the curve $c2 = \rho d^2$ is zero only if $\beta' = 0$. We have

$$\beta' = \frac{2x_i(d-x_i)}{\rho d^3} .$$

Clearly, $\beta' \neq 0$ in either subregion II or subregion III, so we may conclude that there are no critical points in subregion II or subregion III.

We have isolated the possible local minima to the curve

$$c2 = (d-x_i)^2 ,$$

and the ray

$$d = x_i,\ c2 < 0 .$$

Along the curve $c2 = (d-x_i)^2$ we have

$$r(\lambda_i, d, c2) = \frac{|d-x_i|}{d + (d^2 - (d-x_i)^2)^{\frac{1}{2}}} .$$

If we take the directional derivative along this curve, we find

$$r' = \begin{cases} \dfrac{-\left(d + (d^2 - (d-x_i)^2)^{\frac{1}{2}}\right) + (d-x_i)\left(1 + \dfrac{x_i}{(d^2-(d-x_i)^2)^{\frac{1}{2}}}\right)}{\left(d + (d^2 - (d-x_i)^2)^{\frac{1}{2}}\right)^2} & \text{for } d < x_i, \\[2em] \dfrac{\left(d + (d^2 - (d-x_i)^2)^{\frac{1}{2}}\right) - (d-x_i)\left(1 + \dfrac{x_i}{(d^2-(d-x_i)^2)^{\frac{1}{2}}}\right)}{\left(d + (d^2 - (d-x_i)^2)^{\frac{1}{2}}\right)^2} & \text{for } d > x_i, \end{cases}$$

$$= \begin{cases} \dfrac{-x_i}{(d + (d^2 - (d-x_i)^2)^{\frac{1}{2}})(d^2 - (d-x_i)^2)^{\frac{1}{2}}} & \text{for } d < x_i, \\[2ex] \dfrac{x_i}{(d + (d^2 - (d-x_i)^2)^{\frac{1}{2}})(d^2 - (d-x_i)^2)^{\frac{1}{2}}} & \text{for } d > x_i. \end{cases}$$

Since all of the terms are positive we have

$$r' < 0, \quad \text{for} \quad d < x_i,$$

and

$$r' > 0, \quad \text{for} \quad d > x_i.$$

The only local minimum along the curve $c2 = (d-x_i)^2$ occurs at the point $(d, c2) = (x_i, 0)$.

On the ray $d = x_i$, $c2 \leq 0$ we have

$$r(\lambda_i, d, c2) = \frac{(-c2)^{\frac{1}{2}}}{x_i + (x_i^2 - c2)^{\frac{1}{2}}}.$$

Taking the directional derivative along the ray we have

$$r' = \frac{\frac{1}{2}\dfrac{-1}{(-c2)^{\frac{1}{2}}}(x_i + (x_i^2 - c2)^{\frac{1}{2}}) - \frac{1}{2}(-c2)^{\frac{1}{2}}\dfrac{-1}{(x_i^2 - c2)^{\frac{1}{2}}}}{(x_i + (x_i^2 - c2)^{\frac{1}{2}})^2}$$

$$= -\frac{1}{2}\,\frac{x_i}{(x_i + (x_i^2 - c2)^{\frac{1}{2}})(x_i^2 - c2)^{\frac{1}{2}}(-c2)^{\frac{1}{2}}}.$$

Clearly, we have $r' < 0$ along the ray, so that $r(\lambda_i, d, c2)$ decreases as c2 increases to $c2 = 0$.

We can conclude that the point $(d, c2) = (x_i, 0)$ is the only local minimum of the function $r(\lambda_i, d, c2)$ in the region R. At this point we have $r(\lambda_i, x_i, 0) = 0$, and the theorem is proved.

4.3 Pair-wise Best Point

In this section it will be shown that the locus of points for which two surfaces intersect is a continuous curve through the d,c2-plane. In general, the minimum along this curve can be found in terms of the root of a fifth degree polynomial in a certain interval. The coefficients of this polynomial will be given. It can also be shown that this minimum is the only local minimum along the curve, but it will not be proved here. A special case will be treated in which the minimum can be found in terms of the root of a third degree polynomial.

Consider all points $(d, c2) \in R$ such that

$$r(\lambda_i, d, c2) = r(\lambda_j, d, c2) \ .$$

From Theorem 2.5 we know that λ_i and λ_j are on the same member of the family $\mathcal{F}(d,c)$. Moreover, they both satisfy the equation of the ellipse. If $\lambda_i = x_i + iy_j$ and $\lambda_j = x_j + iy_j$, then we can write

$$\frac{(d-x_i)^2}{a2} + \frac{y_i^2}{a2-c2} = 1 \ ,$$

and

$$\frac{(d-x_j)^2}{a2} + \frac{y_j^2}{a2-c2} = 1 \ ;$$

Suppose $y_i = y_j$; then, we have

$$(d-x_i)^2 = (d-x_j)^2 ,$$

which gives

$$d = \frac{x_j + x_i}{2} .$$

Let

$$A = \frac{x_j - x_i}{2} ,$$

$$B = \frac{x_j + x_i}{2} ,$$

$$S = \frac{y_j - y_i}{2} ,$$

$$T = \frac{y_j + y_i}{2} .$$

We may assume that $x_j > x_i$; then, we have $A > 0$, $B > 0$, $T > 0$. For $S = 0$ we have

$$d = B ,$$

and the ellipse equation becomes

$$\frac{A^2}{a2} + \frac{T^2}{a2-c2} = 1 .$$

Solving for c2, we get

$$c2 = \frac{a2(a2 - (A^2 + T^2))}{(a2 - A^2)} .$$

Suppose that $y_i \neq y_j$, that is, $S \neq 0$. Let $\beta = \frac{a2}{c2}$. The ellipse equations become

$$\frac{(d-x_i)^2}{\beta} + \frac{y_i^2}{\beta-1} = c2 ,$$

and

$$\frac{(d-x_j)^2}{\beta} + \frac{y_j^2}{\beta-1} = c2 .$$

Solving for β, we get

$$\beta = \frac{(x_j-x_i)(2d-(x_j-x_i))}{(x_j-x_i)(2d-(x_j-x_i)) + (y_j-y_i)(y_j+y_i)} .$$

In terms of A, B, S, and T, this is

$$\beta = \frac{(d-B)}{(d - (B + \frac{ST}{A}))} .$$

Substituting this back into the ellipse equation, we get

$$c2 = \frac{(d-x_i)^2}{\beta} + \frac{y_i^2}{\beta-1}$$

$$= \frac{(d - (B-A))^2}{\frac{(d-B)}{(d - (B + \frac{ST}{A}))}} + \frac{(T-S)^2}{\frac{\frac{ST}{A}}{(d - (B + \frac{ST}{A}))}}$$

$$= \frac{(d - (B + \frac{ST}{A}))}{(d-B)} ((d - (B-A))^2 + \frac{A(d-B)(T-S)^2}{ST})$$

$$= \frac{(d - (B + \frac{ST}{A}))(d - (B - A\frac{T}{S}))(d - (B - A\frac{S}{T}))}{(d-B)} .$$

Notice that

$$c2 = \frac{1}{\beta}\,(d - (B - A\frac{T}{S}))(d - (B - A\frac{S}{T}))$$

$$= \frac{c2}{a2}\,(d - (B - A\frac{T}{S}))(d - (B - A\frac{S}{T})) \ ,$$

so that

$$a2 = (d - (B - A\frac{T}{S}))(d - (B - A\frac{S}{T})) \ .$$

For $S \neq 0$ we can express c2 and a2 in terms of d. For $S = 0$ we have $d = B$ and can express c2 in terms of a2. To insure that these parameters describe an ellipse, we must have

$$a2 \geq 0 \ ,$$

and

$$a2 - c2 \geq 0 \ .$$

Lemma 4.5 The above equations for c2 and a2 describe the parameters of an ellipse passing through λ_i and λ_j if

$$d < B \quad \text{when} \quad S < 0 \ ,$$

$$d > B \quad \text{when} \quad S > 0 \ ,$$

and

$$d = B,\ a2 \geq A^2 \quad \text{when} \quad S = 0 \ .$$

Proof We have seen that $d = B$ when $S = 0$. For $S = 0$ we have

$$a2 - c2 = a2 - \frac{a2(a2 - (A^2 + T^2))}{(a2 - A^2)}$$

$$= \frac{T^2}{a2 - A^2} \ .$$

For a2 $\geq A^2$ we have a2 - c2 $\geq$ 0. Notice that as a2 approaches A^2, c2 approaches $-\infty$ and when a2 gets large positively, c2 also gets large positively; thus, c2 takes on all possible values.

When S $\neq$ 0 we have

$$c2 = a2 \frac{(d - (B + \frac{ST}{A}))}{(d-B)} ,$$

so that

$$a2 - c2 = a2 - a2 \frac{(d - (B + \frac{ST}{A}))}{(d-B)}$$

$$= a2 \frac{\frac{ST}{A}}{(d-B)} .$$

If S < 0 and a2 $\geq$ 0, then we must have (d-B) < 0. If S > 0 and a2 $\geq$ 0, then we must have (d-B) > 0.

For a2 we have

$$a2 = (d - (B - A\tfrac{T}{S}))(d - (B - A\tfrac{S}{T})) .$$

Since A, B, T, > 0, then if S > 0 and d > B, we have

$$(d-B) + A\frac{T}{S} > 0 ,$$

$$(d-B) + A\frac{S}{T} > 0 ,$$

so a2 > 0. If S < 0 and d < B, then we have

$$(d-B) + A\frac{T}{S} < 0 ,$$

$$(d-B) + A\frac{T}{S} < 0 ,$$

so a2 > 0. This proves the lemma.

A geometric interpretation of this result is shown in Figure 4.5. If $S > 0$, then $y_j > y_i$ and any ellipse through λ_i and λ_j must have its center, d, closer to x_j than x_i, or $B < d$. A similar explanation holds for $S < 0$ and $S = 0$. Figure 4.6 shows what the locus of points in the d,c2-plane might look like in each of the three cases.

Our object is to find the minimum of $r(\lambda_i, d, c2) = r(\lambda_j, d, c2)$ along the curves shown in Figure 4.6. Theorem 4.6 will treat the case $S \neq 0$, and Theorem 4.7 will treat the case $S = 0$.

The following notation will be convenient. Let

$$z = d - B;$$

then, we can write

$$c2 = \frac{(z + A\frac{T}{S})(z + A\frac{S}{T})(z - \frac{ST}{A})}{z} ,$$

$$a2 = (z + A\frac{T}{S})(z + A\frac{S}{T}) .$$

The parameters d, c2, and a2 describe an ellipse through λ_i and λ_j if

$$z > 0 \quad \text{for} \quad S > 0 ,$$

$$z < 0 \quad \text{for} \quad S < 0 .$$

<u>Theorem 4.6</u> If $S \neq 0$, the point at which $r(\lambda_i, d, c2)$ is minimal along the locus of points for which $r(\lambda_i, d, c2) = r(\lambda_j, d, c2)$ can be found as the root of the polynomial

$$p_1 z^5 + p_2 z^4 + p_3 z^3 + p_4 z^2 + p_5 z + p_6 = 0$$

in the interval

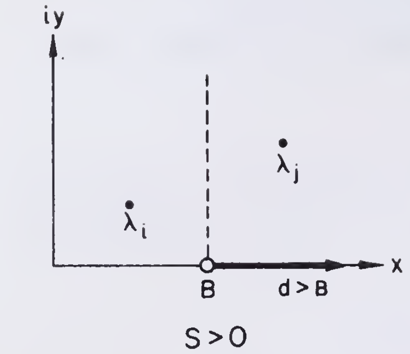

Figure 4.5

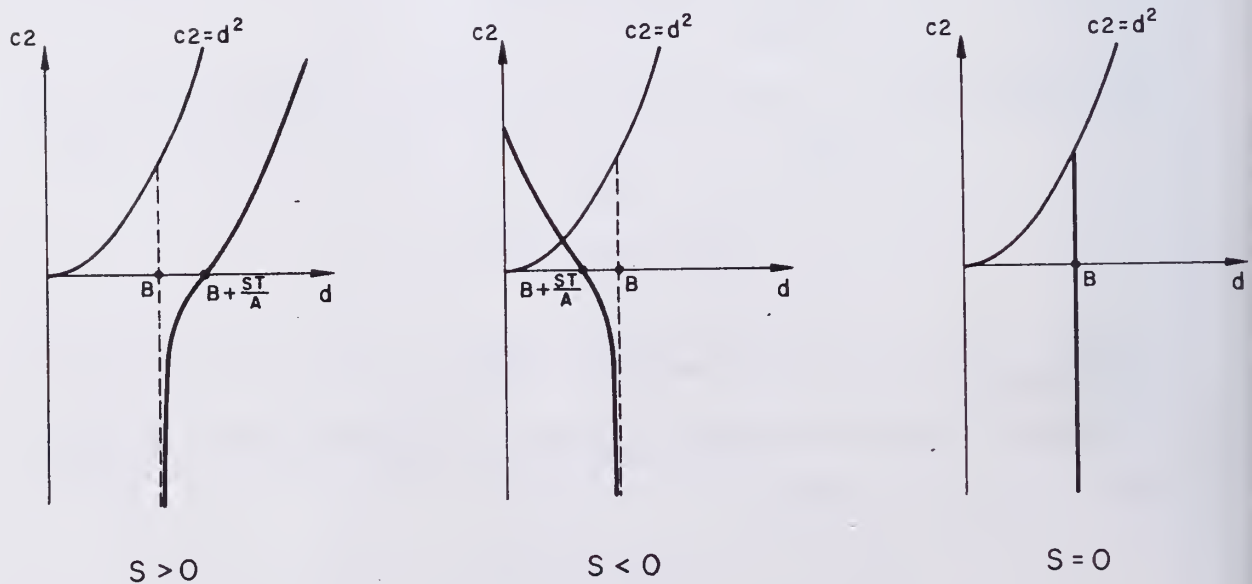

Figure 4.6

$$(0,A) \quad \text{for} \quad S > 0 ,$$
$$(-A,0) \quad \text{for} \quad S < 0 .$$

The coefficients are:

$$p_1 = (2B - A(\frac{T}{S} + \frac{S}{T}))(2B + \frac{ST}{A} - A(\frac{T}{S} + \frac{S}{T})) ,$$

$$p_2 = (2B + \frac{ST}{A} - A(\frac{T}{S} + \frac{S}{T}))((2AB + ST)(\frac{T}{S} + \frac{S}{T}) + 4A^2)$$
$$+ B^2(2B - A(\frac{T}{S} + \frac{S}{T})) + B(B^2 - A^2) ,$$

$$p_3 = 4A^4 - 4A^3B(\frac{T}{S} + \frac{S}{T}) + A^2ST((\frac{T^3}{S^3} + \frac{S^3}{T^3}) - 3(\frac{T}{S} + \frac{S}{T}))$$
$$+ A^2B^2(\frac{T^2}{S^2} + \frac{S^2}{T^2} + 3) ,$$

$$p_4 = AST((B - A\frac{T}{S})(B - 3A\frac{T}{S}) + (B - A\frac{S}{T})(B - 3A\frac{S}{T})) ,$$

$$p_5 = -3A^3ST(2B - A(\frac{T}{S} + \frac{S}{T})) ,$$

$$p_6 = -3A^3ST(B^2 - A^2) .$$

<u>Proof</u> We will refer to $r(\lambda_i, d, c2)$ as $r(z)$. We can write

$$r(z) = \frac{((z + A\frac{T}{S})(z + A\frac{S}{T}))^{\frac{1}{2}} + ((z + A\frac{T}{S})(z + A\frac{S}{T})(1 - \frac{(z - \frac{ST}{A})}{z}))^{\frac{1}{2}}}{(z+B) + ((z+B)^2 - \frac{(z + A\frac{T}{S})(z + A\frac{S}{T})(z - \frac{ST}{A})}{z})^{\frac{1}{2}}} .$$

Since A, B, T > 0, then r(z) is a continuously differentiable function

of z if $z > 0$ when $S > 0$ and $z < 0$ when $S < 0$. Let $\beta = \frac{a2}{c2}$, $\gamma = \frac{d^2}{c2}$ for $c2 \neq 0$. We can write

$$r(z) = \begin{cases} \dfrac{(\beta)^{\frac{1}{2}} + (\beta-1)^{\frac{1}{2}}}{(\gamma)^{\frac{1}{2}} + (\gamma-1)^{\frac{1}{2}}} & \text{for } c2 > 0\ , \\[2ex] \dfrac{(-\beta)^{\frac{1}{2}} + (1-\beta)^{\frac{1}{2}}}{(-\gamma)^{\frac{1}{2}} + (1-\gamma)^{\frac{1}{2}}} & \text{for } c2 < 0\ , \end{cases}$$

with a removable singularity at $c2 = 0$. Taking the derivative with respect to z we have for $c2 > 0$

$$r'(z) = \frac{\left(\frac{1}{2}\,\frac{\beta'}{(\beta)^{\frac{1}{2}}} + \frac{1}{2}\,\frac{\beta'}{(\beta-1)^{\frac{1}{2}}}\right)\left((\gamma)^{\frac{1}{2}} + (\gamma-1)^{\frac{1}{2}}\right) - \left(\frac{1}{2}\,\frac{\gamma'}{(\gamma)^{\frac{1}{2}}} + \frac{1}{2}\,\frac{\gamma'}{(\gamma-1)^{\frac{1}{2}}}\right)\left((\beta)^{\frac{1}{2}} + (\beta-1)^{\frac{1}{2}}\right)}{\left((\gamma)^{\frac{1}{2}} + (\gamma-1)^{\frac{1}{2}}\right)^2}$$

$$= \frac{1}{2}\,\frac{(\beta)^{\frac{1}{2}} + (\beta-1)^{\frac{1}{2}}}{(\gamma)^{\frac{1}{2}} + (\gamma-1)^{\frac{1}{2}}}\left(\frac{\beta'}{(\beta)^{\frac{1}{2}}(\beta-1)^{\frac{1}{2}}} - \frac{\gamma'}{(\gamma)^{\frac{1}{2}}(\gamma-1)^{\frac{1}{2}}}\right)$$

$$= \frac{r(z)}{2}\left(\frac{\beta'}{(\beta)^{\frac{1}{2}}(\beta-1)^{\frac{1}{2}}} - \frac{\gamma'}{(\gamma)^{\frac{1}{2}}(\gamma-1)^{\frac{1}{2}}}\right),$$

and for $c2 < 0$ we have

$$r'(z) = \frac{r(z)}{2}\left(\frac{-\beta'}{(-\beta)^{\frac{1}{2}}(1-\beta)^{\frac{1}{2}}} - \frac{-\gamma'}{(-\gamma)^{\frac{1}{2}}(1-\gamma)^{\frac{1}{2}}}\right)\ ,$$

with a removable singularity at $c2 = 0$. In terms of z we have

$$\beta = \frac{z}{(z - \frac{ST}{A})} ,$$

$$\gamma = \frac{(z+B)^2 z}{(z + A\frac{T}{S})(z + A\frac{S}{T})(z - \frac{ST}{A})} ,$$

$$\beta\text{-}1 = \frac{\frac{ST}{A}}{(z - \frac{ST}{A})} ,$$

$$\gamma\text{-}1 = \frac{(z+B)^2 z - (z + A\frac{T}{S})(z + A\frac{S}{T})(z - \frac{ST}{A})}{(z + A\frac{T}{S})(z + A\frac{S}{T})(z - \frac{ST}{A})}$$

$$= \frac{Q(z)}{(z + A\frac{T}{S})(z + A\frac{S}{T})(z - \frac{ST}{A})} ,$$

where $Q(z)$ is a quadratic. Taking derivatives, we have

$$\beta' = \frac{-\frac{ST}{A}}{(z - \frac{ST}{A})^2} ,$$

$$\gamma' = \frac{((z+B)^2 + 2(z+b)z)(z + A\frac{T}{S})(z + A\frac{S}{T})(z - \frac{ST}{A})}{((z + A\frac{T}{S})(z + A\frac{S}{T})(z + \frac{ST}{A}))^2}$$

$$\frac{- z(z+B)^2((z + A\frac{T}{S})(z + A\frac{S}{T}) + (z + A\frac{T}{S})(z - \frac{ST}{A}) + (z + A\frac{S}{T})(z - \frac{ST}{A}))}{}$$

$$= \frac{C(z)(z+B)}{(z + A\frac{T}{S})(z + A\frac{S}{T})(z - \frac{ST}{A})} ,$$

where c(z) is a cubic. From here on we will assume $S > 0$. The arguments are identical for $S < 0$. Plugging the above expressions into the equation for r'(z), we get

$$r'(z) = -\frac{r(z)}{2}\,\frac{1}{(z - \frac{ST}{A})(z)^{\frac{1}{2}}}\left((\frac{ST}{A})^{\frac{1}{2}} + \frac{C(z)}{(Q(z))^{\frac{1}{2}}(z + A\frac{T}{S})(z + A\frac{S}{T})}\right).$$

This expression has a singularity at $z = \frac{ST}{A}$ which corresponds to $c2 = 0$. To see that this is a removable singularity, notice that

$$C(\frac{ST}{A}) = -(\frac{ST}{A})(B + \frac{ST}{A})(\frac{ST}{A} + \frac{AT}{S})(\frac{ST}{A} + \frac{AS}{T}) \ ,$$

$$Q(\frac{ST}{A}) = \frac{ST}{A}(B + \frac{ST}{A})^2 \ ,$$

so that

$$(\frac{ST}{A})^{\frac{1}{2}} + \frac{C(\frac{ST}{A})}{(Q(\frac{ST}{A}))^{\frac{1}{2}}(\frac{ST}{A} + \frac{AT}{S})(\frac{ST}{A} + \frac{AS}{T})} = 0 \ .$$

We want to show that r'(z) has a root in the interval (0,A). At $z = 0$ we have

$$C(0) = -ABST,$$

$$Q(0) = AST \ .$$

Taking the limit, we have

$$\lim_{z\to 0} r'(z) = \frac{r(0)}{2} \cdot \lim_{z\to 0}\left(\frac{1}{(z)^{\frac{1}{2}}}\right)\left(-\frac{(B-A)}{(ST)^{\frac{1}{2}}}\right)$$

$$= -\infty \ .$$

At $z = A$ we have

$$C(A) = \frac{A^2(T+S)^2}{ST}\,(2(A^2-ST) - A(A+B)) \;,$$

$$Q(A) = \frac{A}{ST}\,(ST(A+B)^2 - (T+S)^2(A^2-ST)) \;,$$

so that

$$r'(A) = -\,\frac{r(A)}{2}\,\frac{(A)^{\frac{1}{2}}}{(A^2-ST)}\left((\frac{ST}{A})^{\frac{1}{2}} + \frac{C(A)ST}{(Q(A))^{\frac{1}{2}}A^2(S+T)^2}\right)$$

$$= -\,\frac{r(A)}{2}\,\frac{(A)^{\frac{1}{2}}}{(A^2-ST)}\left((\frac{ST}{A})^{\frac{1}{2}} + \frac{(2(A^2-ST) - A(A+B))}{(Q(A))^{\frac{1}{2}}}\right)$$

$$= -\,\frac{r(A)}{2}\,\frac{(ST)^{\frac{1}{2}}}{(A^2-ST)}\left(1 + \frac{(2(A^2-ST) - A(A+B))}{(ST(A+B)^2 - (S+T)^2(A^2-ST))^{\frac{1}{2}}}\right) .$$

We would like to show that $r'(A) > 0$. Removing the terms known to be positive, we would like to show that

$$\frac{-1}{(A^2-ST)}\left(1 + \frac{(2(A^2-ST) - A(A+B))}{(ST(A+B)^2 - (S+T)^2(A^2-ST))^{\frac{1}{2}}}\right) > 0 \;,$$

which is equivalent to

$$\frac{A(A+B) - 2(A^2-ST)}{(A^2-ST)} > \frac{(ST(A+B)^2 - (S+T)^2(A^2-ST))^{\frac{1}{2}}}{(A^2-ST)} \;.$$

If $(A^2-ST) > 0$, this is equivalent to

$$A(A+B) - 2(A^2-ST) > (ST(A+B)^2 - (S+T)^2(A^2-ST))^{\frac{1}{2}} \quad .$$

If we square both sides we have the equivalent expression

$$A^2(A+B)^2 - 4A(A+B)(A^2-ST) + 4(A^2-ST)^2 > ST(A+B)^2 - (S+T)^2(A^2-ST) \ ,$$

which is equivalent to

$$(A^2-ST)(4(A^2-ST) - 4A(A+B) + (B+A)^2 + (S+T)^2) > 0 \quad ,$$

which is equivalent to

$$(A^2-ST)((B-A)^2 + (T-S)^2) > 0 \quad ,$$

which is clearly true.

If $(A^2-ST) < 0$, we must show

$$A(A+B) - 2(A^2-ST) < (ST(A+B)^2 - (S+T)^2(A^2-ST))^{\frac{1}{2}} \quad .$$

Since $A(A+B) -2(A^2-ST) > 0$ when $(A^2-ST) < 0$, then squaring both sides and rearranging as before gives the equivalent expression

$$(A^2-ST)((B-A)^2 + (T-S)^2) < 0 \quad ,$$

which is clearly true.

Since $r'(0) = -\infty$, and $r'(A) > 0$, and $r(z)$ is continuously differentiable, we can conclude that $r(z)$ has a minimum in the interval $(0,A)$. At this point we must have $r'(z) = 0$. From the expression for $r'(z)$ we must have

$$(\frac{ST}{A})^{\frac{1}{2}} + \frac{C(z)}{(Q(z))^{\frac{1}{2}}(z + A\frac{T}{S})(z + A\frac{S}{T}} = 0 \quad ,$$

or

$$(\frac{ST}{A})^{\frac{1}{2}}(Q(z))^{\frac{1}{2}} = \frac{-C(z)}{(z + A\frac{T}{S})(z + A\frac{S}{T})} \quad .$$

Squaring both sides, we must have

$$\frac{ST}{A} Q(z)(z + A\frac{T}{S})(z + A\frac{S}{T}) = C(z)^2 \quad .$$

Since $C(z)$ is a cubic and $Q(z)$ is a quadratic, this gives rise to a sixth degree polynomial. One of the roots of this polynomial is at $z = \frac{ST}{A}$, which corresponds to $c2 = 0$, an extraneous root. Factoring out this root, we get the fifth degree polynomial whose coefficients were given in the hypothesis. This polynomial must have a root at the point $z \in (0,A)$ at which $r'(z) = 0$. For $S < 0$ the interval is $(-A,0)$. This completes the proof of the theorem.

It can be shown with arguments based on the derivatives of the fifth degree polynomial that this is the only local minimum of $r(z)$.

We next turn our attention to the case $S = 0$. The following notation will be convenient. Let

$$y = a2;$$

then, we can write

$$d = B ,$$

$$c2 = \frac{y(y - (A^2 + T^2))}{(y-A^2)} \quad ,$$

$$a2 - c2 = \frac{yT^2}{(y-A^2)} \quad .$$

The parameters d, c2, and a2 describe an ellipse through the points λ_i and λ_j if $y > A^2$.

Theorem 4.7 If $S = 0$, the point at which $r(\lambda_i, d, c2)$ is minimal along the locus of points for which $r(\lambda_i, d, c2) = r(\lambda_j, d, c2)$ can be found as the root of the polynomial

$$q_1 y^3 + q_2 y^2 + q_3 y + q_4 = 0$$

in the interval (A^2, B^2). The coefficients are:

$$q_1 = (B^2 + T^2) \ ,$$

$$q_2 = -3A^2B^2 \ ,$$

$$q_3 = 3A^4B^2 \ ,$$

$$q_4 = -A^4B^2(A^2 + T^2) \ .$$

Proof We will refer to $r(\lambda_i, d, c2)$ as $r(y)$. We can write

$$r(y) = \frac{(y)^{\frac{1}{2}} + \left(\frac{yT^2}{y-A^2}\right)^{\frac{1}{2}}}{B + \left(B^2 - \frac{y(y-(A^2+T^2))}{(y-A^2)}\right)^{\frac{1}{2}}} \quad .$$

Notice that for $y \in (A^2, B^2)$, $r(y)$ is a continuously differentiable function of y. If, as in the proof of Theorem 4.6, we let $\beta = \frac{a2}{c2}$, $\gamma = \frac{d2}{c2}$ for $c2 \neq 0$, we have

$$r'(y) = \begin{cases} \dfrac{r(y)}{2}\left(\dfrac{\beta'}{(\beta)^{\frac{1}{2}}(\beta-1)^{\frac{1}{2}}} - \dfrac{\gamma'}{(\gamma)^{\frac{1}{2}}(\gamma-1)^{\frac{1}{2}}}\right) & \text{for } c2 > 0 , \\ \\ \dfrac{r(y)}{2}\left(\dfrac{-\beta'}{(-\beta)^{\frac{1}{2}}(1-\beta)^{\frac{1}{2}}} - \dfrac{-\gamma'}{(-\gamma)^{\frac{1}{2}}(1-\gamma)^{\frac{1}{2}}}\right) & \text{for } c2 > 0 , \end{cases}$$

with a removable singularity at $c2 = 0$. In terms of y we have

$$\beta = \frac{(y-A^2)}{(y-(A^2+T^2))} ,$$

$$\gamma = \frac{B^2(y-A^2)}{y(y-(A^2+T^2))} ,$$

$$\beta-1 = \frac{T^2}{(y-(A^2+T^2))} ,$$

$$\gamma-1 = \frac{-(y^2-(A^2+B^2+T^2)y+A^2B^2)}{y(y-(A^2+T^2))} .$$

Taking the derivative we have

$$\beta' = \frac{-T^2}{(y-(A^2+T^2))^2} ,$$

$$\gamma' = \frac{B^2(y^2-2A^2y+A^2(A^2+T^2))}{(y(y-(A^2+T^2)))^2} .$$

Combining the equations above, we have

$$r'(y) = \frac{r(y)}{2} \frac{1}{(y-(A^2+T^2))(y-A^2)^{\frac{1}{2}}} \left(\frac{B(y^2-2A^2y+A^2(A^2+T^2))}{y(-(y^2-(A^2+B^2+T^2)y+A^2B^2))^{\frac{1}{2}}} - T \right) .$$

We would like to show that $r'(y)$ has a root in the interval (A^2, B^2). We can write

$$\lim_{y\to A^2} r'(y) = \frac{r(A^2)}{2} \cdot \lim_{y\to A^2} \frac{1}{(y-A^2)^{\frac{1}{2}}} \cdot \left(-\left(\frac{B-A}{A}\right)\right)$$

$$= -\infty .$$

If we let $y = B^2$, we have

$$r'(B^2) = \frac{r(B^2)}{2} \frac{1}{(B^2-(A^2+T^2))(B^2-A^2)^{\frac{1}{2}}} \left(\frac{B((B^2-A^2)^2+A^2T^2}{B^2(B^2T^2)^{\frac{1}{2}}} - T \right)$$

$$= \frac{r(B^2)}{2} \frac{1}{(B^2-(A^2+T^2))(B^2-A^2)^{\frac{1}{2}}} \left(\frac{(B^2-A^2)(B^2-(A^2+T^2))}{B^2T} \right)$$

$$= \frac{r(B^2)}{2} \frac{(B^2-A^2)^{\frac{1}{2}}}{B^2T} > 0 .$$

Since $\lim_{y\to A^2} r'(y) = -\infty$, $r'(B^2) > 0$, and $r'(y)$ is continuously differentiable in the interval (A^2, B^2), then $r(y)$ must have a minimum in

the interval (A^2, B^2). From the equation for $r'(y)$ we see that if $r'(y) = 0$ we have

$$B(y^2 - 2A^2 y + A^2(A^2 + T^2)) = T\, y(-(y^2 - (A^2+B^2+T^2)y + A^2B^2))^{\frac{1}{2}} \quad .$$

If we square both sides of this equation and rearrange terms, we have a fourth degree polynomial in y. This polynomial has a root at $y = A^2 + T^2$, which corresponds to $c2 = 0$. Factoring out this extraneous root we have the third degree polynomial whose coefficients were given in the hypothesis.

Any point y such that $r'(y) = 0$ will be a root of this polynomial. If

$$Q(y) = q_1 y^3 + q_2 y^2 + q_3 y + q_4 \ ,$$

then

$$Q'(y) = 3q_1 y^2 + 2q_2 y + q_3 \ .$$

Taking the discriminant of $Q'(y)$, we have

$$4q_2^2 - 12q_1q_3 = 36A^4B^4 - 36A^4B^2(B^2 + T^2)$$

$$= -36A^4B^2T^2 < 0 \quad .$$

We may conclude that $Q'(z)$ has no real roots; thus, $Q(z)$ has only one real root, the one known to give the minimum of $r(y)$. This root is the only local minimum of $r(y)$, and the theorem is proved.

The point at which the minimum occurs will be referred to as the pair-wise best point, and the associated ellipse through λ_i and λ_j will be referred to as the pair-wise best ellipse.

4.4 Three-way Point

If the functions $r(\lambda_i, d, c2)$, $r(\lambda_j, d, c2)$, and $r(\lambda_k, d, c2)$ take on the same value at some point $(d, c2) \in R$, then there is an ellipse, a member of the family $\mathcal{F}(d,c)$, passing through λ_i, λ_j, and λ_k. In this section it will be shown that if such an ellipse exists, it is unique. An existence criterion and parameters of the ellipse will be found in terms of $\lambda_i = x_i + iy_i$, $\lambda_j = x_j + iy_j$, and $\lambda_k = x_k + iy_k$. We will assume that $0 < x_i < x_j < x_k$ and $0 < y_i, y_j, y_k$.

Lemma 4.8 If there exists an ellipse symmetric with respect to the real axis passing through the distinct points λ_i, λ_j, and λ_k, then it is unique. (A general ellipse is determined by 5 points.)

Proof The general equation for an ellipse symmetric with respect to the real axis is

$$Ax^2 + By^2 + Cx = 1 .$$

Suppose the system

$$\begin{pmatrix} x_i^2 & y_i^2 & x_i \\ x_j^2 & y_j^2 & x_j \\ x_k^2 & y_k^2 & x_k \end{pmatrix} \begin{pmatrix} A \\ B \\ C \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

has more than one solution, say

$$\begin{pmatrix} A_1 \\ B_1 \\ C_1 \end{pmatrix} \text{ and } \begin{pmatrix} A_2 \\ B_2 \\ C_2 \end{pmatrix} .$$

If

$$\begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} A_1 \\ B_1 \\ C_1 \end{pmatrix} - \begin{pmatrix} A_2 \\ B_2 \\ C_2 \end{pmatrix} ,$$

then the vector

$$\begin{pmatrix} A \\ B \\ C \end{pmatrix} = \begin{pmatrix} A_1 \\ B_1 \\ C_1 \end{pmatrix} + \omega \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}$$

is a solution for any ω. Suppose $\beta \neq 0$; then, for $\omega = -B_1/\beta$ we have

$$\begin{pmatrix} x_i^2 & y_i^2 & x_i \\ x_j^2 & y_j^2 & x_j \\ x_k^2 & y_k^2 & x_k \end{pmatrix} \begin{pmatrix} A \\ 0 \\ C \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} ,$$

which implies that

$$Az^2 + Cz - 1 = 0$$

has three distinct roots. We can conclude that $\beta = 0$. This in turn implies that

$$\begin{pmatrix} x_i^2 & y_i^2 & x_i \\ x_j^2 & y_j^2 & x_j \\ x_k^2 & y_k^2 & x_k \end{pmatrix} \begin{pmatrix} \alpha \\ 0 \\ \gamma \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} ,$$

or that

$$\alpha z^2 + \gamma z = 0$$

has three distinct roots. We can conclude that $\alpha = \beta = \gamma = 0$, which proves the lemma.

Theorem 4.9 There exists an ellipse symmetric with respect to the real axis passing through λ_i, λ_j, and λ_k if and only if

$$(x_j - x_i)(y_k^2 - y_i^2) < (x_k - x_i)(y_j^2 - y_i^2) \ .$$

If the ellipse exists, it is determined by the parameters

$$d = \frac{1}{2} \frac{(y_i^2(x_j^2 - x_k^2) + y_j^2(x_k^2 - x_i^2) + y_k^2(x_i^2 - x_j^2))}{(y_i^2(x_j - x_k) + y_i^2(x_k - x_i) + y_k^2(x_i - x_j))} \ ,$$

$$a2 = d^2 - \frac{(y_i^2 x_j x_k (x_j - x_k) + y_j^2 x_i x_k (x_k - x_i) + y_k^2 x_i x_j (x_i - x_j))}{(y_i^2(x_j - x_k) + y_j^2(x_k - x_i) + y_k^2(x_i - x_j))} \ ,$$

$$c2 = a2 \left(1 - \frac{(y_i^2(x_j - x_k) + y_j^2(x_k - x_i) + y_k^2(x_i - x_j))}{(x_i - x_j)(x_j - x_k)(x_k - x_i)} \right) \ .$$

Proof If an ellipse is to pass through the three points λ_i, λ_j, and λ_k, then it must be a member of the family of ellipses passing through λ_i and λ_k. Consider the arc of the ellipse connecting λ_i and λ_k as we range over all possible ellipses through λ_i and λ_k (see Figure 4.7). This arc sweeps out the region bounded by $x = x_i$, $x = x_j$, and above the limiting ellipse. For $y_i \neq y_k$, the limiting ellipse is a parabola with the equation

$$x = Py^2 + Q \ ,$$

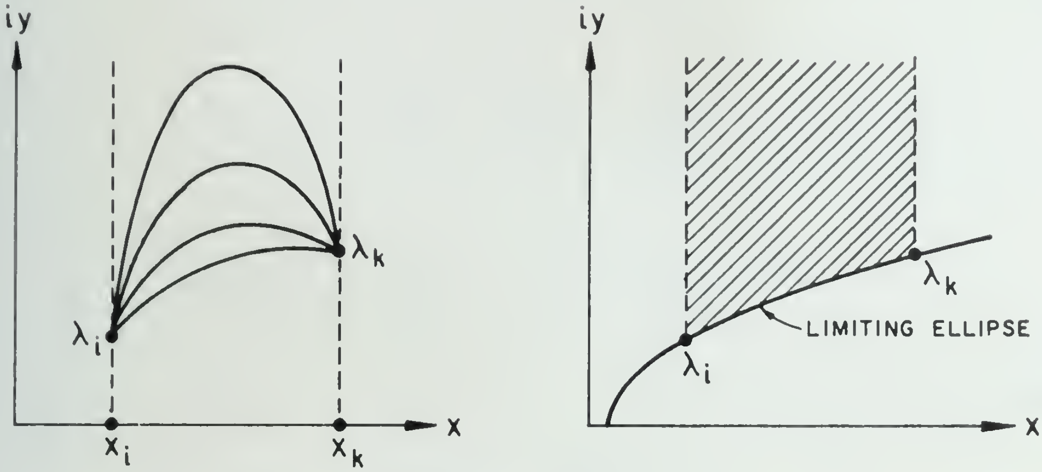

Figure 4.7

where

$$P = \frac{x_k - x_i}{y_k^2 - y_i^2} \quad ,$$

and

$$Q = x_i - \left(\frac{x_k - x_i}{y_k^2 - y_i^2}\right) y_i^2 \quad .$$

For $y_i = y_k$ it is the line

$$y = y_i = y_k \quad .$$

The point λ_j is on one of these ellipses if and only if it is in this region, that is, if and only if

$$x_j < Py_j^2 + Q \quad \text{for} \quad y_k - y_i > 0 \ ,$$

$$x_j > Py_j^2 + Q \quad \text{for} \quad y_k - y_i < 0 \ ,$$

$$y_j > y_i \quad \text{for} \quad y_k - y_k = 0 \ .$$

(See Figure 4.8).

Equivalently, we have

$$(x_j - x_i) < \frac{(x_k - x_i)}{(y_k^2 - y_i^2)} (y_j^2 - y_i^2) \quad \text{for} \quad y_k - y_i > 0 ,$$

$$(x_j - x_i) > \frac{(x_k - x_i)}{(y_k^2 - y_i^2)} (y_j^2 - y_i^2) \quad \text{for} \quad y_k - y_i < 0 ,$$

$$0 < (y_j - y_i) \quad \text{for} \quad y_k - y_i = 0 ,$$

which can be written

$$(x_j - x_i)(y_k^2 - y_i^2) < (x_k - x_i)(y_j^2 - y_i^2)$$

for all cases.

To find the ellipse parameters d, c2, and a2 for an ellipse through λ_i, λ_j, and λ_k, we require that they satisfy the three constraint equations:

1) $$\frac{(d - x_i)^2}{a2} + \frac{y_i^2}{a2 - c2} = 1 ,$$

2) $$\frac{(d - x_j)^2}{a2} + \frac{y_j^2}{a2 - c2} = 1 ,$$

3) $$\frac{(d - x_k)^2}{a2} + \frac{y_k^2}{a2 - c2} = 1 .$$

Assume that λ_i, λ_j, and λ_k satisfy the existence criterion. Then, λ_j must lie in the region described above (see Figure 4.8). We can establish three cases:

Case I. If $y_k - y_i > 0$, then $y_j - y_i > 0$.

Case II. If $y_k - y_i < 0$, then $y_j - y_k > 0$.

Case III. If $y_k - y_i = 0$, then $y_j - y_k > 0$,

$y_j - y_i > 0$.

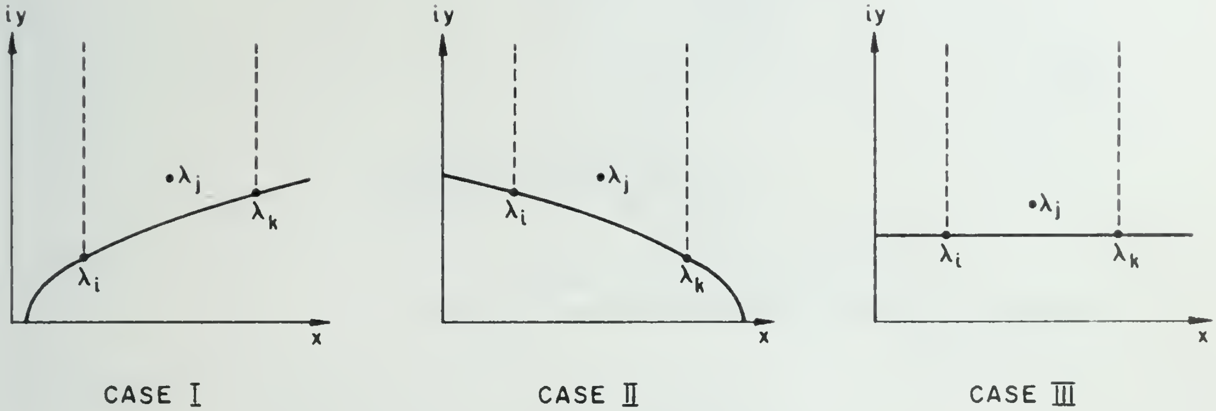

Figure 4.8

Case I. We know from Section 4.3 that if $y_k - y_i \neq 0$, we can use constraint equations 1) and 3) to get an expression for a2 in terms of d. We have

$$a2 = \left(d - \left(\frac{x_k+x_i}{2} - \frac{x_k-x_i}{2}\,\frac{y_k+y_i}{y_k-y_i}\right)\right)\left(d - \left(\frac{x_k+x_i}{2} - \frac{x_k-x_i}{2}\,\frac{y_k-y_i}{y_k+y_i}\right)\right)$$

$$= (d - P_{13})(d - Q_{13}) \;.$$

Likewise, since $y_j - y_i \neq 0$, we can use constraint equations 1) and 2)

to get

$$a2 = \left(d - \left(\frac{x_j+x_i}{2} - \frac{x_j-x_i}{2}\,\frac{y_j+y_i}{y_j-y_i}\right)\right)\left(d - \left(\frac{x_j+x_i}{2} - \frac{x_j-x_i}{2}\,\frac{y_j-y_i}{y_j+y_i}\right)\right)$$

$$= (d - P_{12})(d - Q_{12}) \ .$$

Setting the two right-hand sides equal and solving for d, we get

$$(d - P_{13})(d - Q_{13}) = (d - P_{12})(d - Q_{12}) \ ,$$

and

$$d = \frac{P_{13}Q_{13} - P_{12}Q_{12}}{(P_{13}+Q_{13}) - (P_{12}+Q_{12})} \ .$$

Plugging in the proper expressions, we get

$$d = \frac{1}{2}\,\frac{y_i^2(x_j^2-x_k^2) + y_j^2(x_k^2-x_i^2) + y_k^2(x_i^2-x_j^2)}{y_i^2(x_j-x_k) + y_j^2(x_k-x_i) + y_k^2(x_i-x_j)} \ .$$

(Notice that a positive denominator is equivalent to the existence criterion.)

If we write $d = \frac{1}{2}\,\frac{J}{K}$ and plug this back into the equation

$$a2 = (d - P_{13})(d - Q_{13}) = d^2 - (P_{13} + Q_{13})d + P_{13}Q_{13} \ ,$$

we get

$$a2 = d^2 - \frac{1}{2}\,(P_{13} + Q_{13})\,\frac{J}{K} + P_{13}Q_{13}$$

$$= d^2 - \left(\frac{(P_{13} + Q_{13})J - 2P_{13}Q_{13}K}{2K}\right) \ .$$

Inserting the proper expressions and rearranging terms, we get

$$a2 = d^2 - \frac{y_i^2 x_j x_k (x_j - x_k) + y_j^2 x_i x_k (x_k - x_i) + y_k^2 x_i x_j (x_i - x_j)}{y_i^2 (x_j - x_k) + y_j^2 (x_k - x_i) + y_k^2 (x_i - x_j)}$$

From Section 4.3 we know that since $y_k \neq y_i$ we can also use constraint equations 1) and 3) to get

$$c2 = a2 \frac{\left(d - \left(\frac{x_k + x_i}{2} + \frac{1}{2} \frac{y_k^2 - y_i^2}{x_k - x_i}\right)\right)}{\left(d - \frac{x_k + x_i}{2}\right)}$$

$$= a2 \left(1 - \frac{(y_k^2 - y_i^2)}{(2d(x_k - x_i) - (x_k^2 - x_i^2))}\right) .$$

Again using $d = \frac{1}{2} \frac{J}{K}$, we have

$$c2 = a2 \left(1 - \frac{(y_k^2 - y_i^2)K}{((x_k - x_i)J - (x_k^2 - x_i^2)K)}\right) ,$$

which yields

$$c2 = a2 \left(1 - \frac{y_i^2 (x_j - x_k) + y_j^2 (x_k - x_i) + y_k^2 (x_i - x_j)}{(x_i - x_j)(x_j - x_k)(x_k - x_i)}\right) .$$

Case II. In this case $y_k - y_i \neq 0$ and $y_j - y_k \neq 0$, so we may use constraint equations 1) and 3), and 2) and 3) to get expressions

$$a2 = (d - P_{13})(d - Q_{13}) ,$$

and

$$a2 = (d - P_{23})(d - Q_{23}) \ .$$

Using the same procedure as in Case I, we arrive at the same expressions for d, a2, and c2.

Case III. In this case $y_j - y_i \neq 0$ and $y_j - y_k \neq 0$. Using constraint equations 1) and 2), and 2) and 3), we again achieve the same results. This proves the theorem.

The point at which three functions take on the same value will be referred to as a three-way point, and the associated ellipse through λ_i, λ_j, and λ_k will be referred to as a three-way ellipse.

4.5 The Algorithm

The results of the first four sections of this chapter have shown that the solution of the mini-max problem,

$$\min_{(d,c2)\epsilon R} \ \max_{\lambda_i \epsilon H^+} \ r(\lambda_i, d, c2) \ ,$$

is the minimum point of a single function, a pairwise best point, or a three-way point. In this section an algorithm will be developed to find the mini-max solution among the possible candidates.

From the Alternative Theorem we know that if a point is the mini-max solution, the function or functions it is associated with must equal the maximum over all functions at that point. Equivalently, the associated ellipse must contain the positive hull, H^+, in the closure of its interior. For this reason the minimum point of a single function cannot be the mini-max solution.

Lemma 4.10 If the positive hull, H^+, contains more than one eigenvalue, then the mini-max solution cannot be the minimum point of a single function.

Proof From the Alternative Theorem we know that the mini-max solution may occur at a local minimum of a single function, say $r(\lambda_j,d,c2)$, only if

$$r(\lambda_j,d,c2) = \max_{\lambda_i \in H^+} r(\lambda_i,d,c2)$$

at that point. That is to say, the associated ellipse through λ_j contains the positive hull in the closure of its interior. From Theorem 4.3 we know that the only local minimum of $r(\lambda_j,d,c2)$ occurs at $(d,c2) = (x_j,-y_j^2)$. The associated ellipse is the degenerate ellipse with foci at λ_j and $\bar{\lambda}_j$. If H^+ contains more than one eigenvalue, say $\lambda_k \neq \lambda_j$, then λ_k must lie outside the degenerate ellipse. Thus, we have

$$r(\lambda_k,x_j,-y_j^2) > r(\lambda_j,x_j,-y_j^2) \ ,$$

which proves the lemma.

We now know that the mini-max solution is either a pair-wise best point or a three-way point.

Theorem 4.11 A pair-wise best point whose associated ellipse contains the positive hull, H^+, in the closure of its interior is the mini-max solution. If no such pair-wise best point exists, then the mini-max solution can be found among the three-way points whose associated ellipses contain the positive hull.

Proof Suppose the pair-wise best point associated with λ_j and λ_k, call it $(d_{jk},c2_{jk})$, is such that the associated ellipse passing through λ_j and λ_k contains H^+. Then, we have

$$r(\lambda_j, d_{jk}, c2_{jk}) = \max_{\lambda_i \epsilon H^+} r(\lambda_i, d_{jk}, c2_{jk})$$

$$\geq \min_{(d,c2)\epsilon R} \max_{\lambda_i \epsilon H^+} r(\lambda_i, d, c2)$$

$$\geq \min_{(d,c2)\epsilon R} \max\{r(\lambda_j, d, c2), r(\lambda_k, d, c2)\}$$

$$= r(\lambda_j, d_{jk}, c2_{jk}) \; ,$$

which proves the first result. The last equality is a result of Lemma 4.10 and the Alternative Theorem. The mini-max problem over two functions must have a pair-wise best point for its solution.

Suppose that no pair-wise best point qualifies. From the Alternative Theorem and Lemma 4.10, we know that the mini-max solution is a three-way point whose associated ellipse contains the positive hull. This proves the theorem.

To find the mini-max solution, one must systematically take each pair of eigenvalues in the positive hull and find the pair-wise best point. If the associated ellipse contains the positive hull, then the point is the mini-max solution. If no pair-wise best point qualifies, one must take each combination of three eigenvalues and look for a three-way point. If it exists and its associated ellipse contains the positive hull, the point is a candidate. The mini-max solution is the three-way candidate that yields the smallest convergence factor, $r(\lambda, d, c2)$.

Notice that those eigenvalues in H^+ which are involved in some combination of three eigenvalues that produced a three-way candidate

are exactly the key elements described in Section 3.2. In the course of finding the mini-max solution, the key elements are determined. Thus, we may discard those eigenvalues in H^+ which are not key elements. Reducing the number of eigenvalues in H^+ reduces the number of combinations of three that must be tried in subsequent searches for a mini-max solution.

5. ADAPTIVE PROCEDURE

The algorithm for finding optimal iteration parameters developed in Chapter 2 and Chapter 3 depends upon knowledge of the spectrum of the matrix A. In practice, however, little is known about the spectrum of A. In this chapter a procedure will be developed for estimating the hull of the spectrum from information acquired during the iteration, allowing dynamic improvement of the iteration parameters. Section 5.1 will show how eigenvalue estimates can be extracted from the sequence of residuals. Section 5.2 will show how these estimates can be used to build an approximate hull of the eigenvalues of A.

5.1 Modified Power Method

Suppose that the iteration parameters d and c2 have been chosen from some prior knowledge of the matrix A. After n steps of the Stiefel Iteration based upon d and c2, the error can be expressed as

$$e_n = P_n(A)e_0 \ .$$

Multiplying this expression by A, we can write the residual as

$$r_n = P_n(A)r_0 \ .$$

Suppose that r_0 can be written as a linear combination of the eigenvectors of A; then, we have

$$r_0 = \sum_{i=1}^{k} (\alpha_i v_i + \overline{\alpha}_i \overline{v}_i) \ ,$$

where v_i is the eigenvector of A associated with the eigenvalue λ_i and $\|v_i\| = 1$. Since A is a real valued matrix and r_0 is a real valued vector, the eigenvalues and eigenvectors appear in complex conjugate pairs. After n steps of the iteration, we have

$$r_n = P_n(A)r_0 = \sum_{i=1}^{k} (\alpha_i P_n(\lambda_i)v_i + \bar{\alpha}_i P_n(\bar{\lambda}_i)\bar{v}_i) .$$

From Section 3.1 we know that

$$P_n(\lambda) \doteq (M(\lambda))^n$$

for large n. Let

$$m_i = M(\lambda_i)$$

be the eigenvalue of the operator M(A) corresponding to the eigenvalue λ_i of A. For large n we can write

$$r_n \doteq \sum_{i=1}^{k} (\alpha_i m_i^n v_i + \bar{\alpha}_i \bar{m}_i^n \bar{v}_i) .$$

Suppose m_d and m_s are the dominant and subdominant eigenvalues of M(A); i.e., suppose

$$|m_d| \geq |m_s| > |m_i| \quad \text{for} \quad i \neq d,s .$$

Then, for large n the residual can be written as

$$r_n = m_d^n(\alpha_d v_d) + \bar{m}_d^n(\bar{\alpha}_d \bar{v}_d) + m_s^n(\alpha_s v_s) + \bar{m}_s^n(\bar{\alpha}_s \bar{v}_s) + \epsilon_n ,$$

where ϵ_n is small relative to the other terms. The residual is nearly a linear combination of the eigenvectors associated with the dominant convergence factors. Likewise, we have

$$r_{n+1} = m_d^{n+1}(\alpha_d v_d) + \overline{m}_d^{n+1}(\overline{\alpha}_d \overline{v}_d) + m_s^{n+1}(\alpha_s v_s) + \overline{m}_s^{n+1}(\overline{\alpha}_s \overline{v}_s) + \epsilon_{n+1} ,$$

$$r_{n+2} = m_d^{n+2}(\alpha_d v_d) + \overline{m}_d^{n+2}(\overline{\alpha}_d \overline{v}_d) + m_s^{n+2}(\alpha_s v_s) + \overline{m}_s^{n+2}(\overline{\alpha}_s \overline{v}_s) + \epsilon_{n+2} ,$$

$$r_{n+3} = m_d^{n+3}(\alpha_d v_d) + \overline{m}_d^{n+3}(\overline{\alpha}_d \overline{v}_d) + m_s^{n+3}(\alpha_s v_s) + \overline{m}_s^{n+3}(\overline{\alpha}_s \overline{v}_s) + \epsilon_{n+3} ,$$

$$r_{n+4} = m_d^{n+4}(\alpha_d v_d) + \overline{m}_d^{n+4}(\overline{\alpha}_d \overline{v}_d) + m_s^{n+4}(\alpha_s v_s) + \overline{m}_s^{n+4}(\overline{\alpha}_s \overline{v}_s) + \epsilon_{n+4} .$$

If we let

$$\begin{aligned} Q(z) &= z^4 + q_1 z^3 + q_2 z^2 + q_3 z + q_4 \\ &= (z-m_d)(z-\overline{m}_d)(z-m_s)(z-\overline{m}_s) , \end{aligned}$$

then, ignoring the ϵ terms, we have

$$\| r_{n+4} + q_1 r_{n+3} + q_2 r_{n+2} + q_3 r_{n+1} + q_4 r_n \| \doteq 0 .$$

Choosing (q_1, q_2, q_3, q_4) to make the ℓ_2-norm of the above expression as small as possible is equivalent to finding the least squares solution of the system

$$\begin{pmatrix} | & | & | & | \\ r_{n+3} & r_{n+2} & r_{n+1} & r_n \\ | & | & | & | \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix} = - \begin{pmatrix} | \\ r_{n+4} \\ | \end{pmatrix} .$$

The roots of the polynomial

$$Q(z) = z^4 + q_1 z^3 + q_2 z^2 + q_3 z + q_4 ,$$

whose coefficients are the solution of the least squares problem above, are approximations to the dominant eigenvalues of the operator $M(A)$.

In light of the asymptotic equivalence of the two operators

$$P_n(A) \doteq (M(A))^n ,$$

this procedure corresponds to the power method for simultaneous estimation of the eigenvalues of $M(A)$ (Wilkinson [24]). The number of estimates produced is determined by the degree of the polynomial $Q(z)$ and the number of previous residual vectors stored. The method can be adjusted to yield any number of eigenvalue estimates.

The accuracy of this approximation depends upon the separation of the eigenvalues of $M(A)$ (Wilkinson [24]). For the purposes here, it is only important that the error lies in the direction of the other eigenvalues of $M(A)$. If A is symmetric, this will hold.

Theorem 5.1 If A is symmetric, the eigenvalue approximations produced from the procedure described above will lie in the convex hull of the eigenvalues of $M(A)$.

Proof If A is symmetric, then it has a complete orthonormal set of eigenvectors. If

$$\begin{aligned} Q(z) &= z^4 + q_1 z^3 + q_2 z^2 + q_3 z + q_4 \\ &= (z-\eta_1)(z-\bar{\eta}_1)(z-\eta_2)(z-\bar{\eta}_2) , \end{aligned}$$

then for large n we have

$$\|r_{n+4} + q_1 r_{n+3} + q_2 r_{n+2} + q_3 r_{n+1} + q_4 r_n\|^2$$

$$= \|\sum_{i=1}^{k} (\alpha_i m_i^n Q(m_i) v_i)\|^2$$

$$= \sum_{i=1}^{k} |\alpha_i|^2 |m_i|^{2n} |Q(m_i)|^2 \quad .$$

If $|Q(m_i)|$ can be made smaller for each m_i, then the entire sum will be smaller. We have

$$|Q(m_i)| = |m_i - \eta_1||m_i - \bar{\eta}_1||m_i - \eta_2||m_i - \bar{\eta}_2| \ .$$

Suppose η_1 lies outside the convex hull of the eigenvalues of M(A) (see Figure 5.1). Both $|m_i - \eta_1|$ and $|m_i - \bar{\eta}_1|$ can be made smaller for

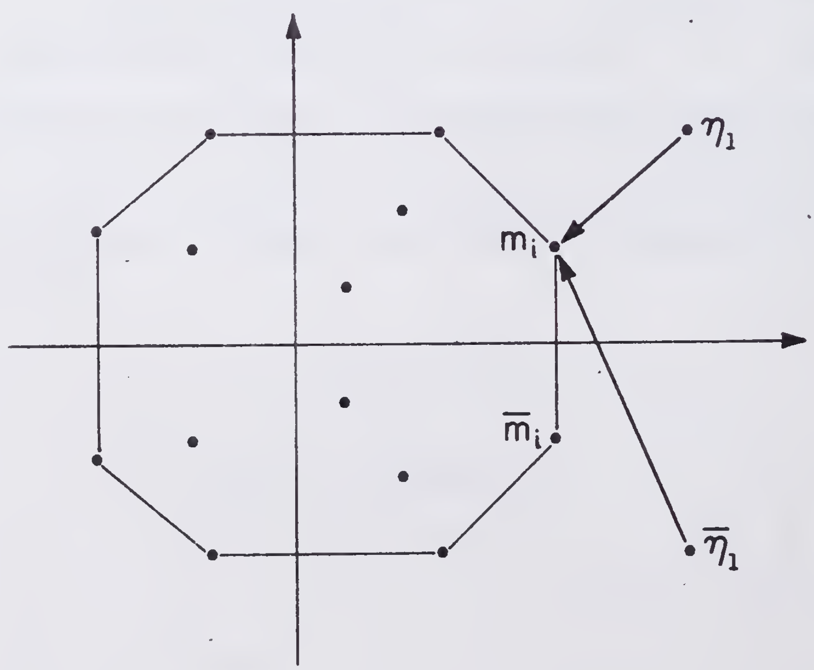

Figure 5.1

each m_i by moving η_1 toward the convex hull of the eigenvalues of $M(A)$. By the same reasoning the sum can be reduced by moving η_2 toward the convex hull. We may conclude that the least squares solution will yield eigenvalue approximations that lie inside the convex hull, which proves the theorem.

No such result is known by the author for a general matrix. Indeed, the loss of orthogonality and the possibility of nonlinear elementary divisors complicate the analysis. Experimental results on large sparse matrices, however, have exhibited the tendency of the approximate eigenvalue to lie inside the convex hull of the true eigenvalues.

The relationship between eigenvalues of A and eigenvalues of $M(A)$ is given by the function

$$M(\lambda) = e^{\left(\cosh^{-1}\left(\frac{d-\lambda}{c}\right) - \cosh^{-1}\left(\frac{d}{c}\right)\right)}$$

$$= \frac{(d-\lambda) + ((d-\lambda)^2 - c2)^{\frac{1}{2}}}{d + (d^2 - c2)^{\frac{1}{2}}} .$$

This function maps the members of the family $\mathcal{F}(d,c)$ onto circles centered at the origin (Section 2.1). The radius of the circle is

$$|M(\lambda)| = r(\lambda, d, c2) ,$$

the convergence factor associated with the ellipse. The region of convergence in the λ-plane, shown in Figure 5.2, is mapped injectively onto the region in the m-plane shown in Figure 5.2. Since the branch

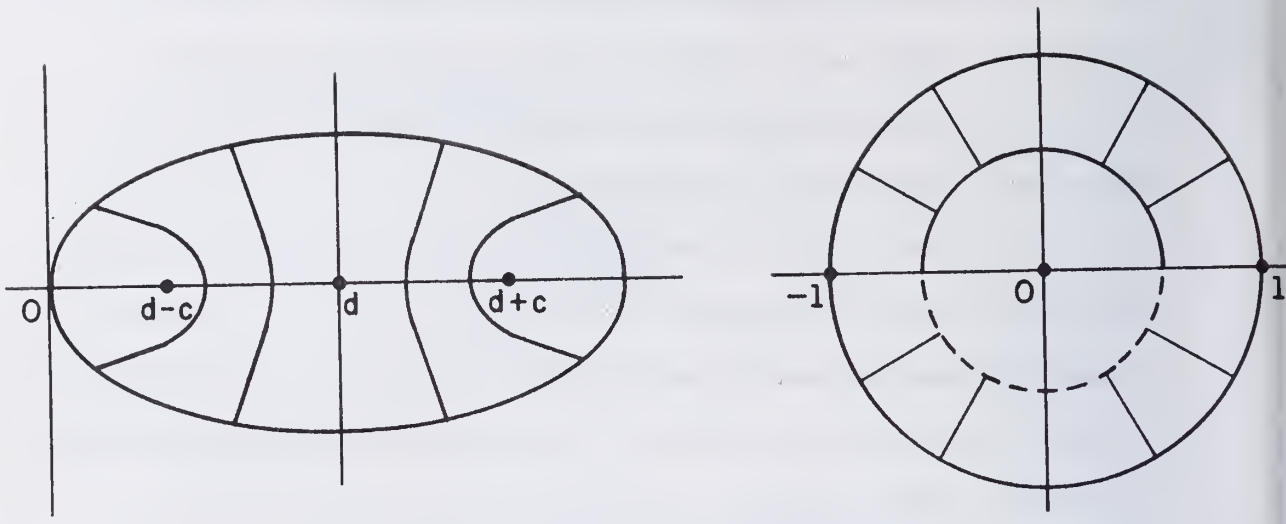

Figure 5.2

of $\cosh^{-1}$ with positive real part is used, we have

$$|M(\lambda)| = |e^{(\cosh^{-1}(\frac{d-\lambda}{c}) - \cosh^{-1}(\frac{d}{c}))}|$$

$$\geq |e^{-\cosh^{-1}(\frac{d}{c})}|$$

$$= \frac{1}{|(\frac{d}{c}) + ((\frac{d}{c})^2 - 1)^{\frac{1}{2}}|}$$

$$= \frac{|c2|^{\frac{1}{2}}}{d + (d-c2)^{\frac{1}{2}}} \quad .$$

Let

$$g = d + (d^2 - c2)^{\frac{1}{2}} \quad ;$$

then, we can write

$$\lambda = d - \frac{1}{2}\left(gm + \frac{c2}{gm}\right) ,$$

with the restriction

$$|m| \geq \frac{|c2|^{\frac{1}{2}}}{g} .$$

If poor separation of the eigenvalues of M(A) causes the modified power method to yield an eigenvalue approximation m such that

$$|m| < \frac{|c2|^{\frac{1}{2}}}{g} ,$$

then it must be discarded.

This map is conformal, but it is not linear. If m lies in the convex hull of $\{m_i\}$, λ does not necessarily lie in the convex hull of $\{\lambda_i\}$. An underestimation of the imaginary part of m_i may cause an overestimation of the real part of $d-\lambda_i$. This warping is slight, however, and shows little effect in practice.

5.2 Procedure and Example

A four step procedure for dynamic improvement of iteration parameters is outlined below.

1. Choose initial d and c2.

The initial choice of d and c2 must be based upon prior knowledge of the matrix A. In Section 1.2, crude bounds were established which gave rectangular regions known to contain the spectrum of A. These regions could be used to choose initial d and c2. If the system were scaled so that each diagonal term of A had the value 1, then d = 1, c2 = 0, representing circles centered at 1, could be used.

2. Iterate.

Perform a predetermined number of steps of the Stiefel iteration based upon d and c2, and store the last five residuals. The number of steps should be large enough to insure separation for the modified power method.

3. Get eigenvalue estimates.

Using the modified power method described in Section 5.1, obtain eigenvalue estimates. Add the new eigenvalue estimates to any previous eigenvalue estimates and form the positive hull.

4. Choose new d and c2.

Using the algorithm outlined in Section 4.5, find the optimal d and c2 for this set of eigenvalue estimates.

Repeat 2, 3, and 4. These steps will be referred to as a <u>cycle</u>.

To illustrate how this procedure might work, suppose the hull of the eigenvalues of A is as shown in Figure 5.3. Suppose d and c2 have been chosen as indicated in Figure 5.4. Several members of $\mathcal{F}(d,c)$ are also shown in Figure 5.4. It is clear that λ_1 and λ_2 are the dominant and subdominant eigenvalues for this choice of d and c2.

After a sufficient number of steps of the Stiefel iteration, the modified power method will give eigenvalue estimates ρ_1 and ρ_2 shown in Figure 5.5.* If we let d+c and d-c also be eigenvalue estimates, then the approximate hull is as shown in Figure 5.5. A choice of d and c2 based upon this approximate hull yields the d and c shown in Figure 5.6. Some members of the new family $\mathcal{F}(d,c)$ are shown in Figure 5.6, and it

*The eigenvalue estimates were chosen for illustrative purposes. The values of d and c2 were computed from these values, and the figures were plotted by the IBM CALCOMP Plotter at the University of Illinois.

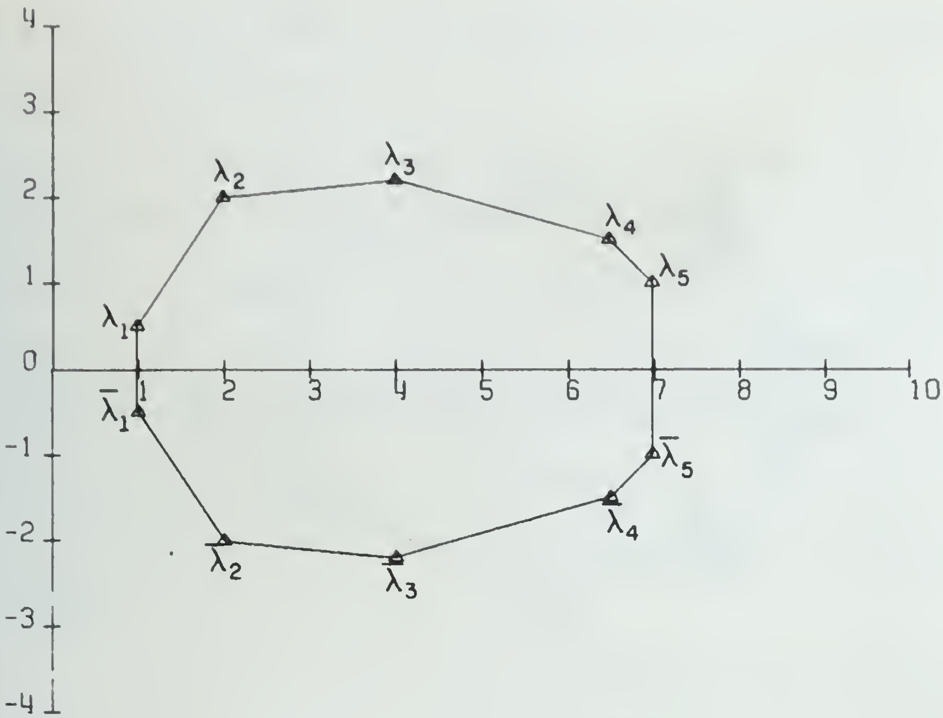

Figure 5.3

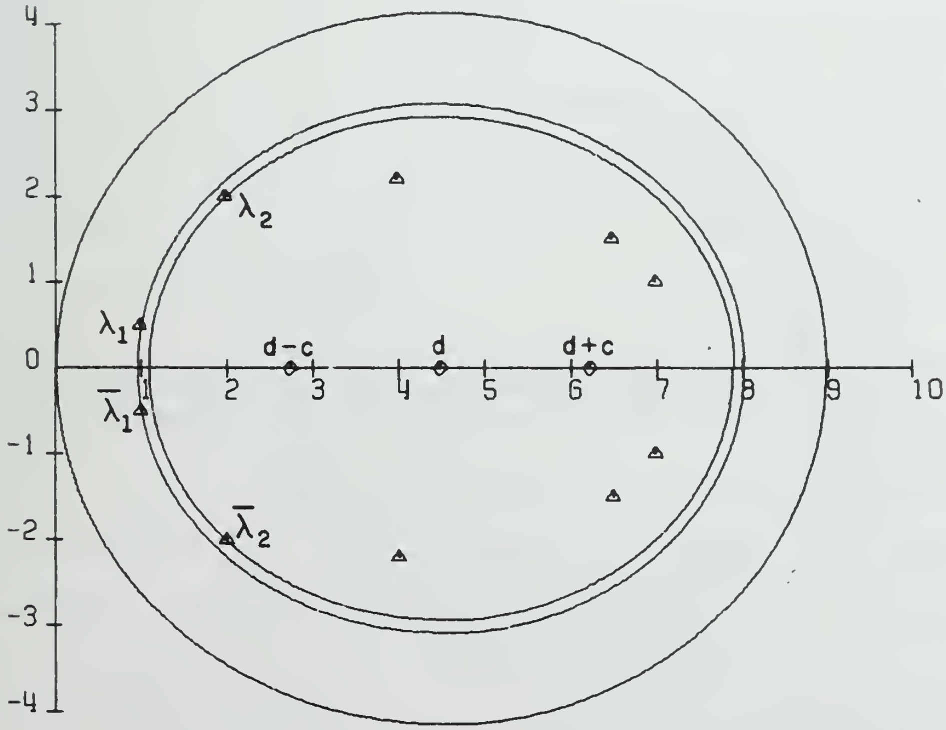

Figure 5.4

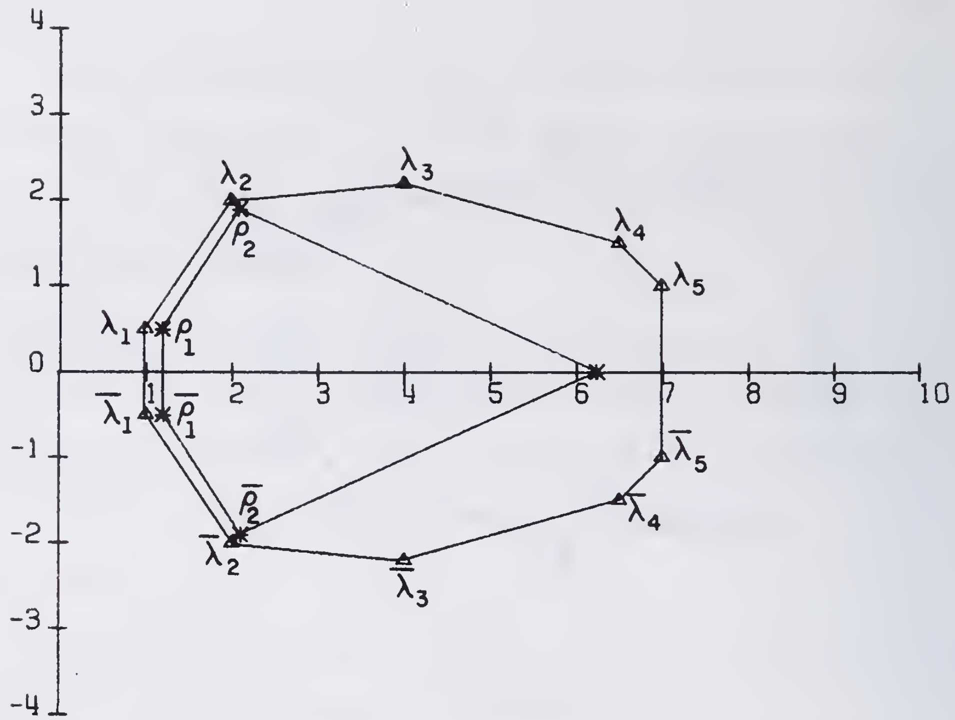

Figure 5.5

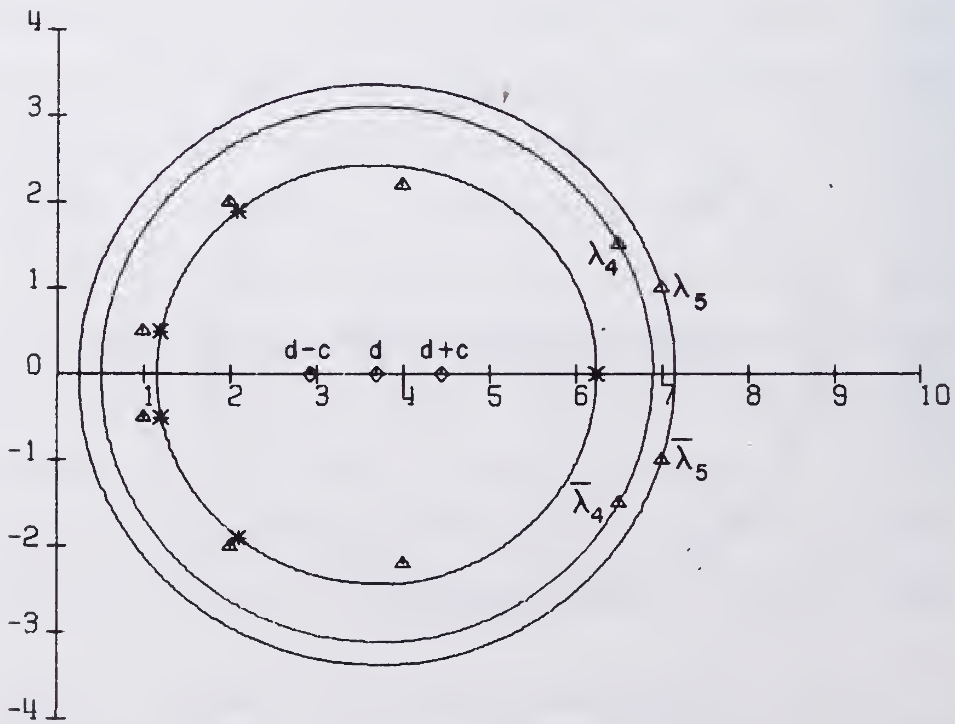

Figure 5.6

is clear that λ_5 and λ_4 are the dominant and subdominant eigenvalues for this choice of d and c2.

After a sufficient number of steps of the Stiefel iteration, the modified power method will give eigenvalue approximations ρ_5 and ρ_4 as shown in Figure 5.7. The new approximate hull, shown in Figure 5.7, describes the true hull fairly well. The optimal choice of d and c2 based upon this approximate hull yields the d and c shown in Figure 5.8. The best ellipse enclosing the approximate hull is labeled "computed" in Figure 5.8, and the best ellipse enclosing the true hull is labeled "optimal". They differ only slightly.

If further eigenvalue estimates fall inside the approximate hull, they will provide no new information and will be ignored. If further eigenvalue estimates lie outside the approximate hull but inside the true hull, they will help to better describe the true hull and will lead to better values of d and c2.

Notice that in this example the true hull was inside the region of convergence, the ellipse passing through the origin, for each choice of d and c2 (see Figure 5.4). This may not always be the case. If an eigenvalue λ_i is outside of the region of convergence, then the corresponding eigenvalue m_i of M(A) will have modulous greater than unity. The error in the direction of the eigenvector associated with λ_i will increase at each step. However, m_i will dominate so strongly that at the end of the cycle, the modified power method will yield an excellent approximation of λ_i. The next choice of d and c2 will involve this estimate. If a provision is made to return to the iterate at the beginning of the cycle, a poor choice of d and c2 will not move the iteration toward convergence, but it will give rise to new iteration parameters that will produce convergence.

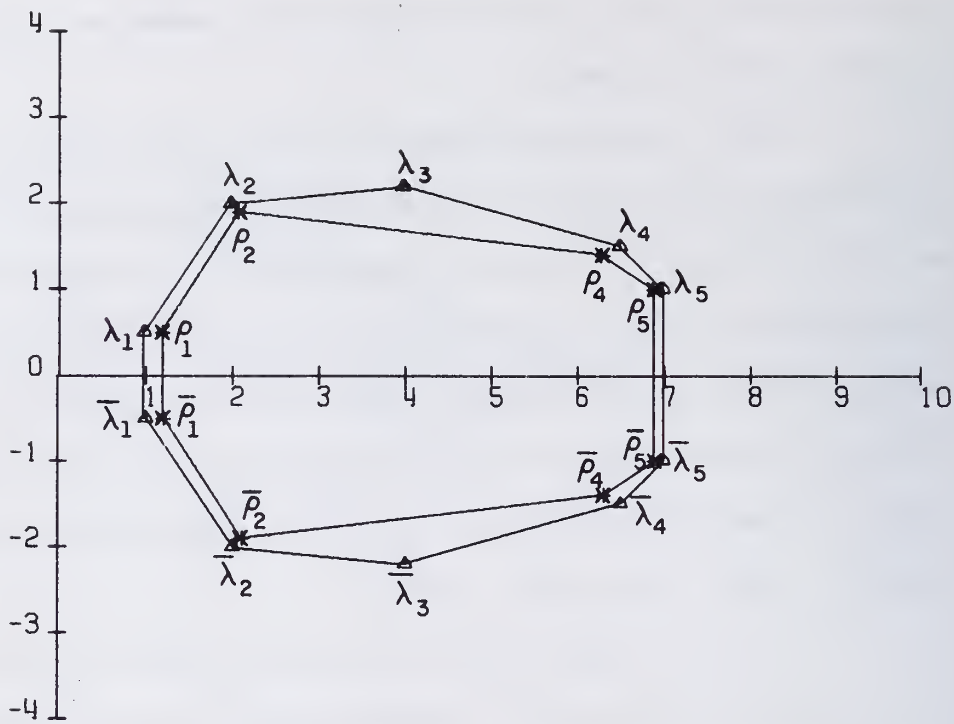

Figure 5.7

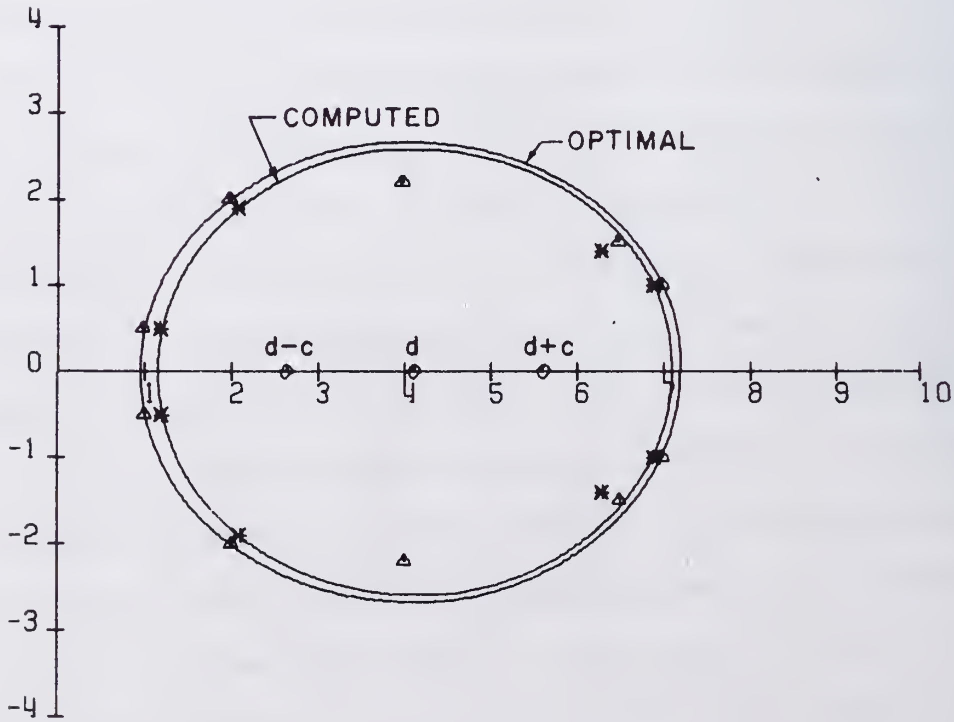

Figure 5.8

6. IMPLEMENTATION AND RESULTS

The implementation of the Tchebychef algorithm will be discussed in Section 6.1, followed by a discussion of competing methods in Section 6.2, and experimental results in Section 6.3. A listing of the subroutines coded in FORTRAN will appear in the appendix.

6.1 Implementation

The Stiefel iteration is performed by the subroutine TCHEB (Appendix A). A basic flowchart of TCHEB appears in Figure 6.1. To solve the system Ax = b, the user must supply the matrix A of dimension N, a storage scheme, and a subroutine to perform matrix vector multiplication (NSYMAX in the listing). The user must also supply an initial guess at the solution x (X(N)) and the target vector b (B(N)). Storage space must be provided for:

R(N)	the residual,
DX(N)	the change in X at each step,
XLAST(N)	X at the start of the cycle,
RLAST(N)*	R at the start of the cycle,
S(N,4)	the four previous residuals at the end of the cycle.

The dimension parameter N must be passed in a common statement as well as the parameter ND, the number of diagonals required to store A.

---

*This vector may be replaced by a matrix vector multiplication in the subroutine LASTX (Appendix A).

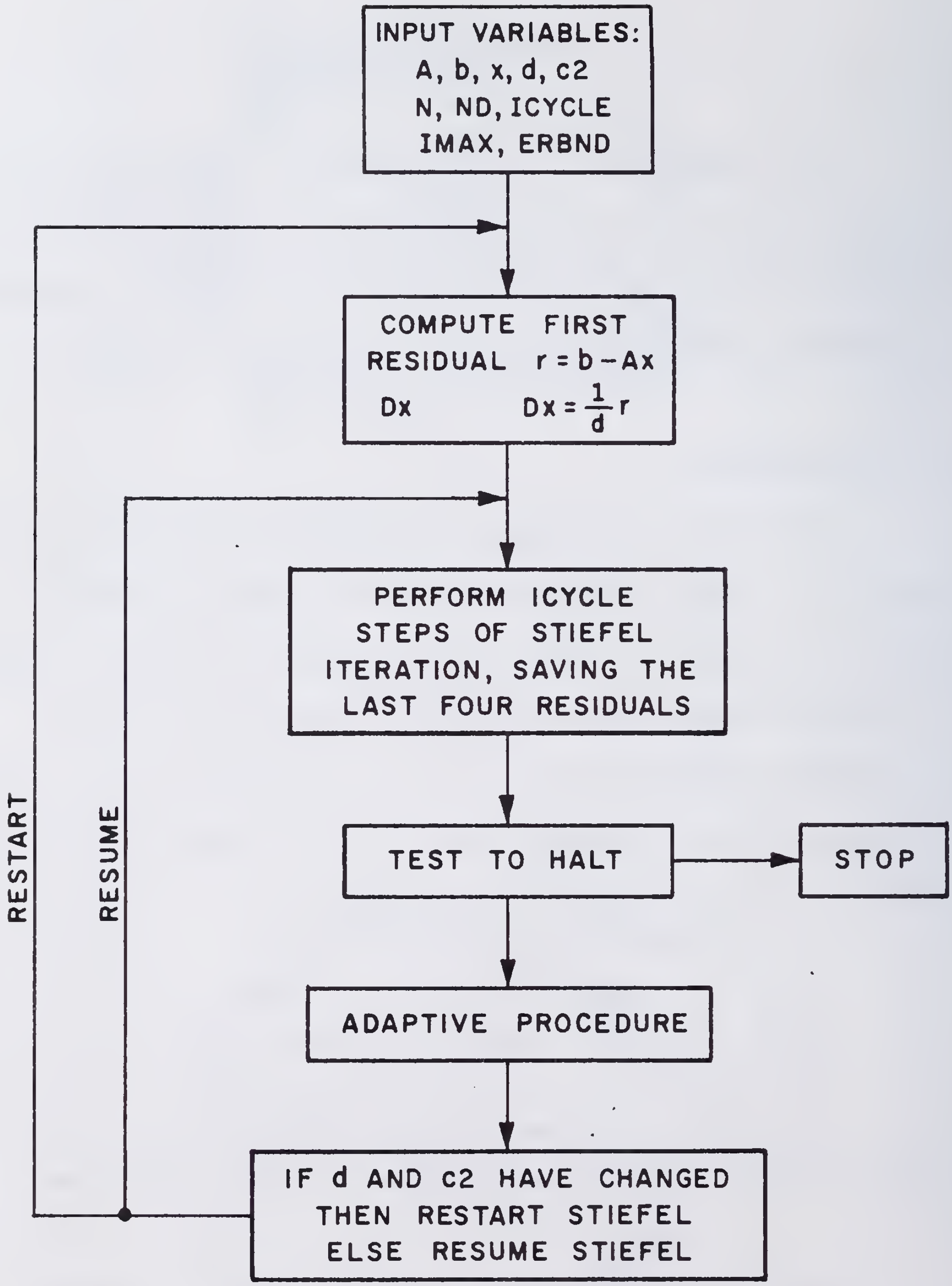
INPUT VARIABLES:
A, b, x, d, c2
N, ND, ICYCLE
IMAX, ERBND
COMPUTE FIRST
RESIDUAL r = b - Ax
Dx Dx = 1/d r
PERFORM ICYCLE
STEPS OF STIEFEL
ITERATION, SAVING THE
LAST FOUR RESIDUALS
TEST TO HALT
STOP
ADAPTIVE PROCEDURE
IF d AND c2 HAVE CHANGED
THEN RESTART STIEFEL
ELSE RESUME STIEFEL
RESTART
RESUME

Figure 6.1

The initial choice of d and c2 must also be passed in a common statement. This choice can be based on rough bounds like those established in Section 1.2. A poor choice of d and c2 will cause the first cycle to be spent in obtaining eigenvalue estimates (Section 5.2). Care should be taken to avoid a choice that causes overflow before the end of the cycle, at which time the program will adjust itself.

The positive hull of the eigenvalue estimates is stored in the doubly linked list CH(25,2) with link array ICH(25,2). The foci d+c and d-c associated with the initial choice of d and c2 are used as initial eigenvalue estimates. For that reason d and c2 should be chosen so that d+c and d-c lie inside the convex hull of the eigenvalues of A. If the user wishes to supply a priori eigenvalues, he must initialize CH and ICH in the subroutine INIT (Appendix A).

The number of iterative steps per cycle must be set by the input parameter ICYCLE. In practice, little difference was found for values of ICYCLE from 15 to 50. The value 20 was used in most of the results that follow.

The other input parameters are IMAX, the maximum number of iterations allowed, and ERBND, the acceptable bound on the ℓ_2-norm of the residual vector. Since the hull of the eigenvalues of A is found by the adaptive procedure, a bound on the modulus of the smallest eigenvalue is available. This may be incorporated in the error bound by replacing ERBND by ERBND*(D-DSQRT(A2)) in the test to halt in subroutine TCHEB.

The adaptive procedure, called from subroutine TCHEB, is broken into four parts (see subroutine ADAPT, Appendix A). The first

part, subroutine LSTSQ, involves finding the least squares solution to the system

$$S_{N\times 4} * Q_{4\times 1} = - R_{N\times 1} ,$$

where the columns of S are the previous residuals (Section 5.1). Subroutine LSTSQ converts this system to the equivalent system

$$S^T S_{4\times 4} * Q_{4\times 1} = - S^T R_{4\times 1} .$$

Since the residuals may become small, this system is multiplied by a constant so that $S^T S(1,1) = 1.0$. The matrix $S^T S$ is symmetric, so it can be computed by taking 10 inner products. Likewise, 4 inner products are required to compute $-S^T R$. The normalized system is, in general, less stable than the large system, but results have shown that this instability does not warrant the extra work involved in solving the large system.

The normalized system may be solved in many ways. If one eigenvalue of M(A) dominates strongly, however, the matrix $S^T S$ may be singular (Section 5.1), in which case a least squares solution is desired. For that reason a Bidiagonalization method with reorthogonalization was used to solve the 4×4 system (Golub and Kahan [11]). The subroutines for the Bidiagonalization method appear in Appendix B.

Subroutine EVS finds the roots of the polynomial associated with the least squares solution Q and converts them into eigenvalue estimates (Section 5.1). Any root finding routine may be used to find the roots of the fourth degree polynomial. A library routine, RSSR, from the University of Illinois FORTUOI Library was used by the author. A listing of RSSR appears in Appendix C.

The subroutine HULL adds the new eigenvalue estimates to the previous eigenvalue estimates and forms the positive hull (Section 3.2). If none of the new eigenvalue estimates are members of the positive hull, control is returned to the subroutine TCHEB, and the iteration is resumed.

The subroutine ELLIP finds the best ellipse enclosing the positive hull (Section 4.5). If the best ellipse is a three-way ellipse, this subroutine also determines the key elements of the positive hull and discards the others (Section 3.2 and Section 4.4). Finding a pair-wise best ellipse involves finding the root of a fifth degree polynomial in a certain interval (Section 4.3). This was done by the routine ZEROIN, which appears in Shampine and Allen [18]. A version adapted for the purposes here appears in Appendix C.

6.2 Competing Algorithms

Two competing iterative methods for solving the nonsymmetric linear system $Ax = b$ are the Bidiagonalization method (Golub and Kahan [11], Paige [17]) and the method of Conjugate Gradients applied to the equivalent system, $A^TAx = A^Tb$ (Hestenes and Stiefel [12]). The Tchebychef algorithm requires more storage than the other two in that four previous residuals must be retained for the adaptive procedure. The other methods require more work per iterative step. Table 6.1 shows a comparison of the work per iterative step and storage requirements of the three methods.

Table 6.1

	MUL/STEP	ADD/STEP	STORAGE
TCHEB	(ND+2.7)N*	(ND+3.7)N	(ND+9)N
BIDIAG	(2ND+7)N	(2ND+5)N	(ND+5)N
CG on A^TA	(2ND+6)N	(2ND+6)N	(ND+5)N

Here N is the dimension of the system, and ND is the number of columns of length N required to store the matrix A. If A is a 5-point difference operator, then ND = 5. Finite element matrices usually require more storage. For instance, when cubic shape functions are used in two dimensions, we may have ND = 33 and in three dimensions we may have ND = 135 (Zienkiewicz [27]). As the storage required for the matrix A becomes greater, the extra storage required by the Tchebychef algorithm becomes less of a factor, but the work required by the other methods remains twice as much as that required by the Tchebychef algorithm.

Both the Bidiagonalization method and the method of Conjugate Gradients on the equivalent system, $A^TAx = A^Tb$, are sensitive to the condition of A^TA, whereas the Tchebychef method is sensitive to the condition of A. Because of this advantage and the advantage of less work per iterative step, the Tchebychef method was considerably faster than the other methods on a series of test problems. In all of the tests that were run, the Bidiagonalization method was slightly better than the method of Conjugate Gradients. For that reason results are shown for the Bidiagonalization method only.

---

*The adaptive procedure in the Tchebychef algorithm requires 14N multiplications and additions (Section 6.1). If an adaptive procedure is carried out every 20 steps (ICYCLE = 20) then the average overhead is .7N per step.

6.3 Results

The convergence properties of the Tchebychef algorithm depend only upon the spectrum of the matrix A and not upon any special zero structure. It is desirable, however, to test the algorithm on easily constructable systems whose spectrums have known properties. Consider the second order linear differential operator on a a rectangular domain:

$$-\frac{\partial}{\partial x}\left(A_1(x,y)\frac{\partial}{\partial x}\right) - \frac{\partial}{\partial y}\left(A_2(x,y)\frac{\partial}{\partial y}\right) + B_1(x,y)\frac{\partial}{\partial x} + B_2(x,y)\frac{\partial}{\partial y} + Q(x,y) ,$$

where the functions A_1, A_2, B_1, B_2, and Q are continuous and differentiable. This operator can be broken into two parts:

$$-\frac{\partial}{\partial x}\left(A_1(x,y)\frac{\partial}{\partial x}\right) - \frac{\partial}{\partial y}\left(A_2(x,y)\frac{\partial}{\partial y}\right) + Q(x,y) ,$$

and

$$B_1(x,y)\frac{\partial}{\partial x} + B_2(x,y)\frac{\partial}{\partial y} .$$

With suitable boundary conditions, the first part is a self-adjoint operator while the second is not (Dunford and Schwartz [4]). Accordingly, if we approximate this operator by a finite difference operator on a regular grid, the first part gives rise to a symmetric matrix, and the second part gives rise to a nonsymmetric matrix. Let h be the grid size and (x_i,y_j) be the grid points with the standard left-to-right, down-to-up ordering of node points. Using central differences and Dirichlet boundary conditions, let A be the 5-point difference matrix associated with the differential operator. The matrix A can be written in two parts: $A = M + N$, corresponding to the two parts of the differential operator. The matrix M is symmetric and, if k is the number of grid points per

column, can be written in block form as

$$M = \begin{pmatrix} M_1 & C_1 & & & 0 \\ C_1 & M_2 & C_2 & & \\ & C_2 & \ddots & \ddots & C_{k-1} \\ 0 & & & C_{k-1} & M_k \end{pmatrix} .$$

The M_i's are tridiagonal and the C_i's are diagonal. If ℓ is the number of gridpoints per row, we have

$$M_i = \begin{pmatrix} a_{1i} & b_{1i} & & & 0 \\ b_{1i} & a_{2i} & b_{2i} & & \\ & b_{2i} & a_{3i} & \ddots & \\ & & \ddots & \ddots & b_{\ell-1i} \\ 0 & & & b_{\ell-1i} & a_{\ell i} \end{pmatrix},$$

and

$$C_i = \begin{pmatrix} c_{1i} & & & 0 \\ & c_{2i} & & \\ & & \ddots & \\ 0 & & & c_{\ell i} \end{pmatrix} ,$$

where

$$a_{i,j} = \frac{1}{h^2}\{A_1(x_i - \frac{h}{2}, y_j) + A_1(x_i + \frac{h}{2}, y_j)$$

$$+ A_2(x_i, y_j - \frac{h}{2}) + A_2(x_i, y_j + \frac{h}{2})\} + Q(x_i, y_j) ,$$

$$b_{ij} = \frac{1}{h^2} A_1(x_i + \frac{h}{2}, y_j) ,$$

$$c_{ij} = \frac{1}{h^2} A_2(x_i, y_j + \frac{h}{2}) .$$

The matrix N is nonsymmetric and can be written in block form as

$$N = \begin{pmatrix} N_1 & F_1 & & & 0 \\ -F_2 & N_2 & F_2 & & \\ & -F_3 & N_3 & \ddots & \\ & & \ddots & \ddots & F_{k-1} \\ 0 & & & -F_k & N_k \end{pmatrix} .$$

Again, the N_i's are tridiagonal and the F_i's are diagonal. We have

$$N_i = \begin{pmatrix} 0 & d_{1i} & & & 0 \\ -d_{2i} & 0 & d_{2i} & & \\ & -d_{3i} & 0 & \ddots & \\ & & \ddots & \ddots & d_{\ell-1i} \\ 0 & & & -d_{\ell i} & 0 \end{pmatrix} ,$$

and

$$F_i = \begin{pmatrix} f_{1i} & & & 0 \\ & f_{2i} & & \\ & & \ddots & \\ 0 & & & f_{\ell i} \end{pmatrix},$$

where

$$d_{ij} = \frac{1}{2h} B_1(x_i, y_j) ,$$

$$f_{ij} = \frac{1}{2h} B_2(x_i, y_j) .$$

a) Nonsingular 5-point Difference Matrices

If $A_1(x,y)$, $A_2(x,y) > 0$ and $Q(x,y) \geq 0$, then M is a positive definite matrix. If $B_1(x,y)$ and $B_2(x,y)$ are constant functions, then N is antisymmetric (Varga [22]). From Section 1.2 we know that the eigenvalues of A = M+N lie in the right half plane. In particular, let

$$A_1(x,y) = A_2(x,y) = 1 ,$$

$$Q(x,y) = 0 ,$$

$$B_1(x,y) = B_2(x,y) = \beta ,$$

on the square region

$$0 \leq x \leq 41, \quad 0 \leq y \leq 41 ,$$

with grid length $h = 1$. Then, A is a 1600×1600 matrix and

$$M_i = \begin{pmatrix} 4 & -1 & & & 0 \\ -1 & 4 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ 0 & & & -1 & 4 \end{pmatrix},$$

$$C_i = \begin{pmatrix} -1 & & 0 \\ & \ddots & \\ 0 & & -1 \end{pmatrix},$$

$$N_i = \begin{pmatrix} 0 & \frac{\beta}{2} & & & 0 \\ -\frac{\beta}{2} & 0 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \frac{\beta}{2} \\ 0 & & & -\frac{\beta}{2} & 0 \end{pmatrix},$$

$$F_i = \begin{pmatrix} \frac{\beta}{2} & & 0 \\ & \ddots & \\ 0 & & \frac{\beta}{2} \end{pmatrix}.$$

The eigenvalues of the positive definite matrix M lie in the open interval (0,8), while the eigenvalues of the antisymmetric matrix N lie in the interval $(-2\beta i, 2\beta i)$ along the imaginary axis. If λ is an eigenvalue of A, it is contained in the rectangular region

$$\mathrm{Re}(\lambda) \in (0,8) \ ,$$

$$\mathrm{Im}(\lambda) \in (-2\beta, 2\beta) \ .$$

The Tchebychef algorithm was tested against the two competing methods for values of β ranging from .1 to 40 (see Table 6.2). Figures 6.2(a) -6.10(a) show the hull of the eigenvalue estimates computed by the Tchebychef algorithm and the best ellipse enclosing the approximate hull. For small β the matrix A is nearly symmetric which yields a hull close to the real axis (see Figure 6.2(a)). For large β the matrix A is nearly antisymmetric which yields a nearly verticle spectrum (see Figure 6.10(a)).

Table 6.2

Figure #	β	Initial d	Initial c	Optimal d	Optimal c	Convergence Factor	Rate of Congergence	ICYCLE
6.2	.1	4.0	3.87	4.69	4.66	.9075	.04215	20
6.3	.4	4.0	3.87	4.01	3.87	.9502	.02217	40
6.4	.8	4.0	0	3.94	3.31	.9737	.01155	20
6.5	2	4.0	0	3.93	3.09i	.9571	.01906	20
6.6	4	4.0	0	4.05	8.86i	.9558	.01964	20
6.7	8	4.0	15i	3.85	18.44i	.9324	.03039	20
6.8	10	4.0	14.14i	4.05	20.18i	.9494	.02254	20
6.9	20	4.0	31.62i	4.24	40.39i	.9595	.02024	20
6.10	40	4.0	75.00i	5.28	85.28i	.9769	.01015	20

Figure 6.2(a) Best ellipse for β = .1

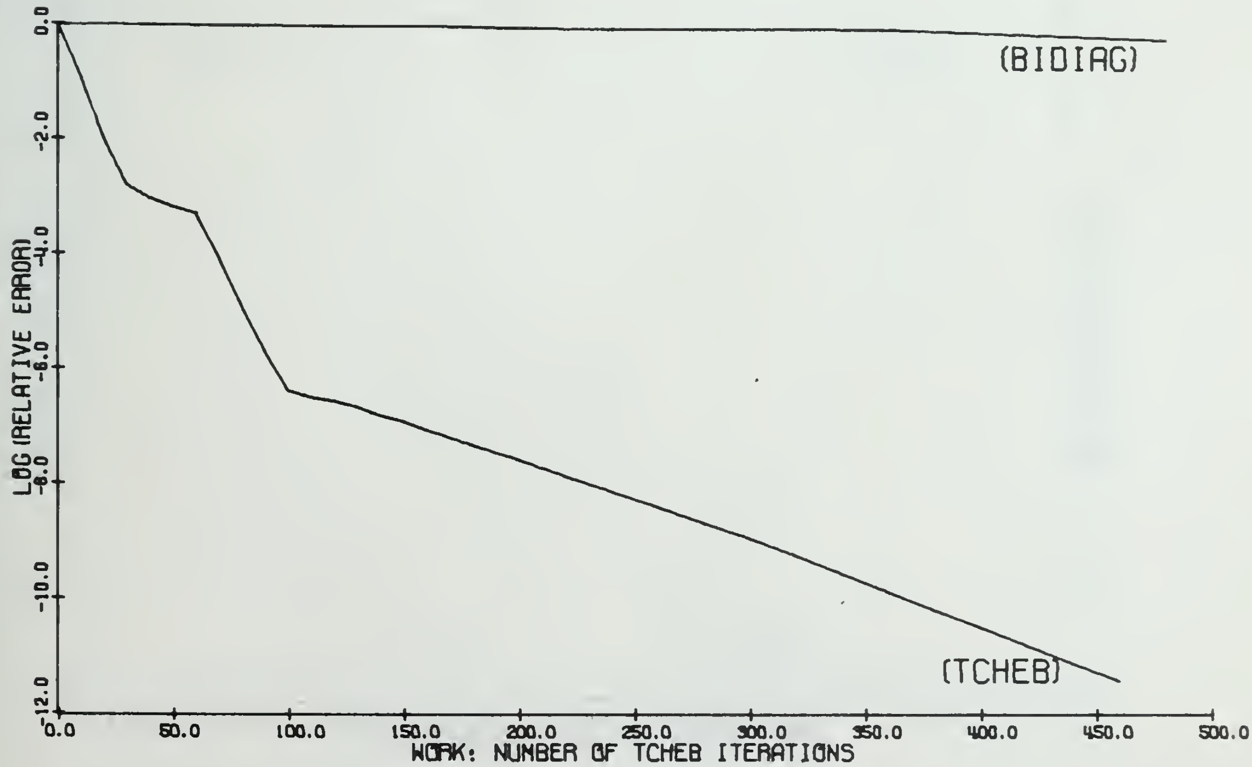

Figure 6.2(b) Work required for β = .1

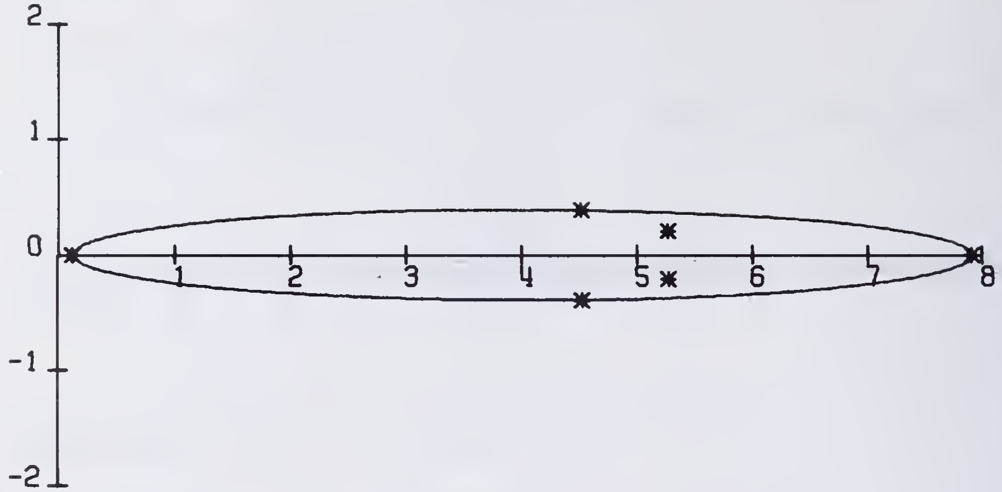

Figure 6.3(a) Best ellipse for β = .4

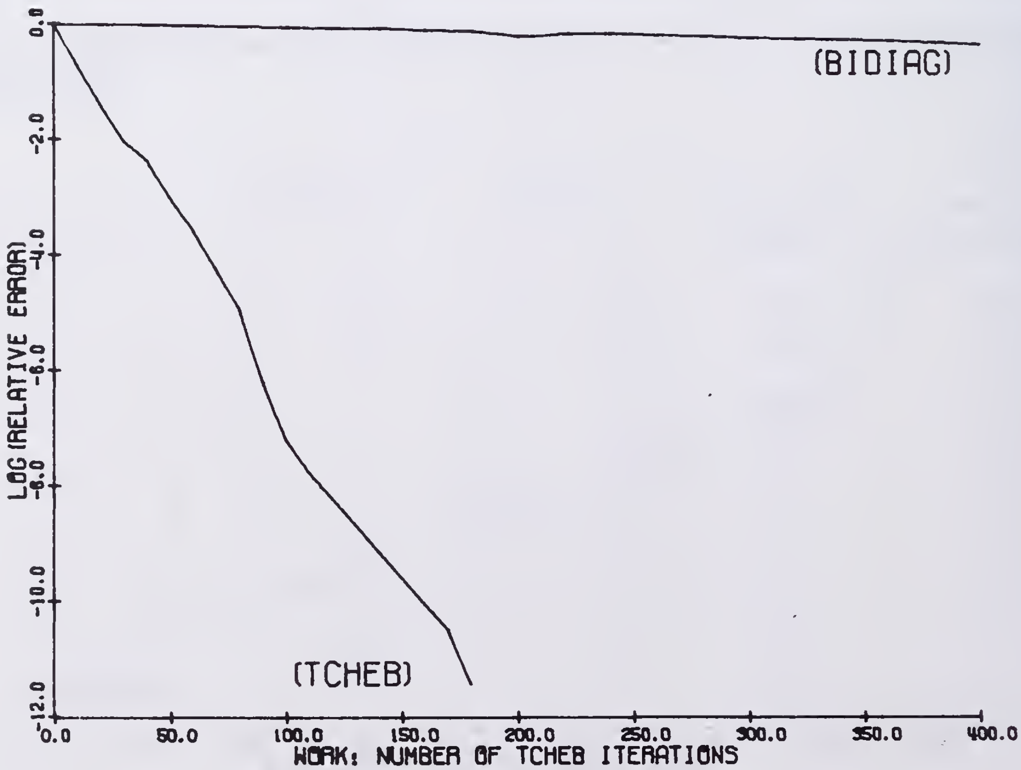

Figure 6.3(b) Work required for β = .4

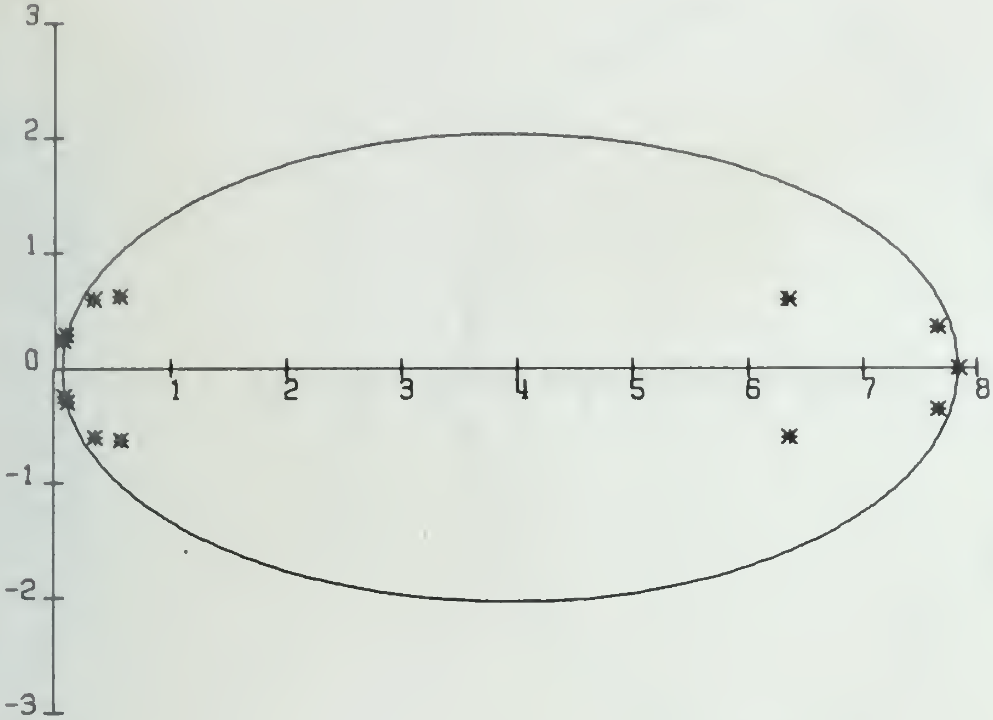

Figure 6.4(a) Best ellipse for β = .8

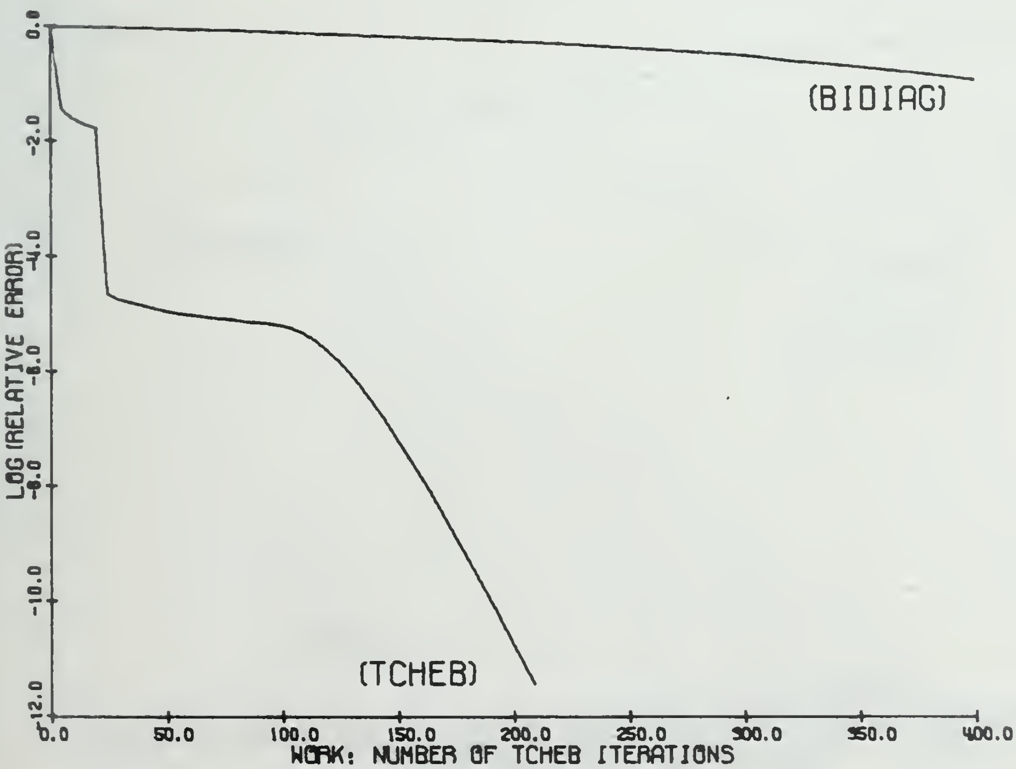

Figure 6.4(b) Work required for β = .8

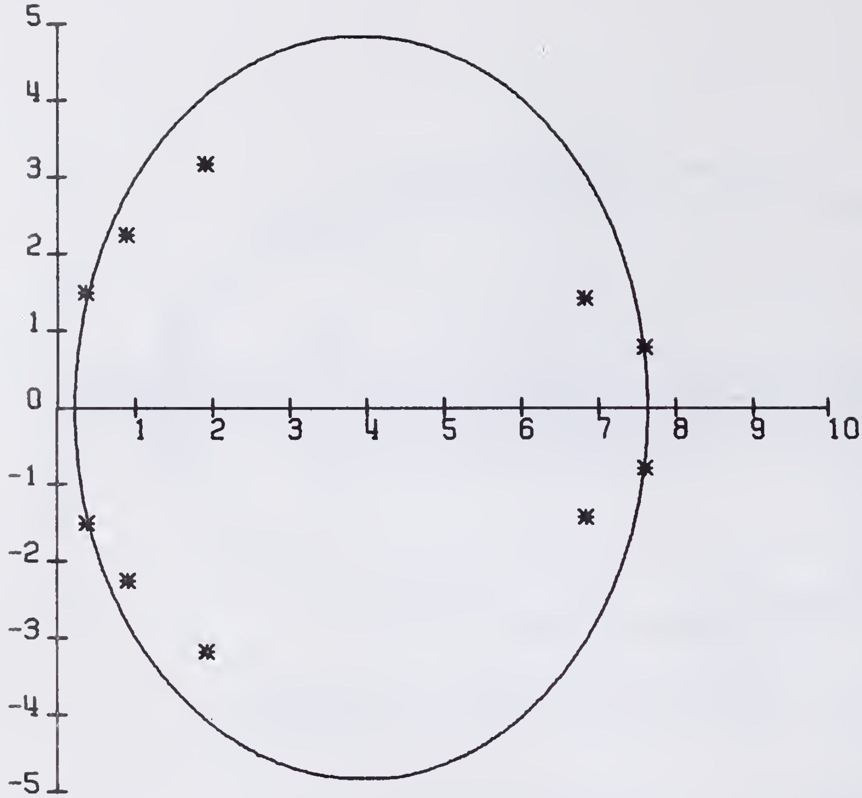

Figure 6.5(a) Best ellipse for $\beta = 2$

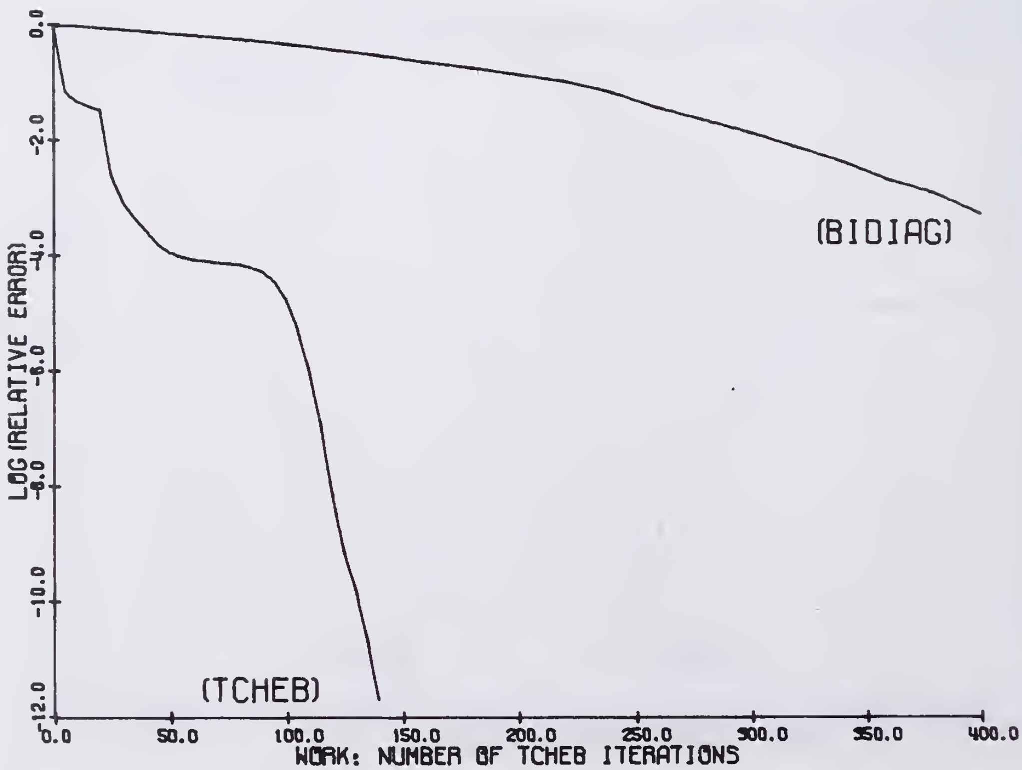

Figure 6.5(b) Work required for $\beta = 2$

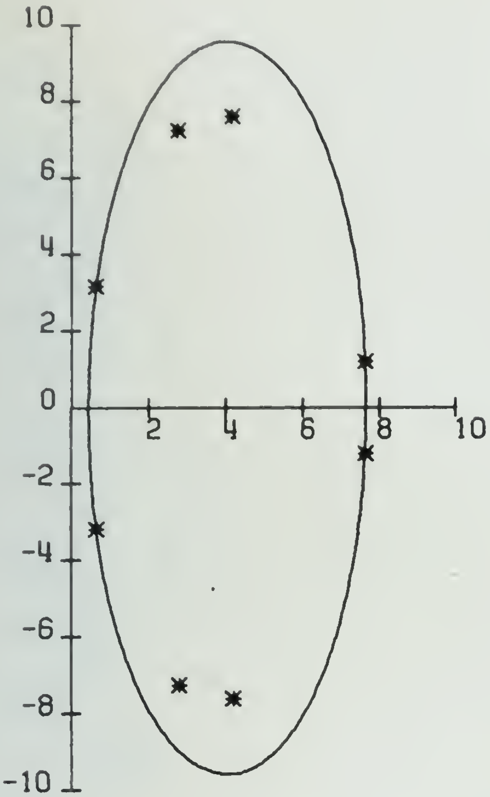

Figure 6.6(a) Best ellipse for $\beta = 4$

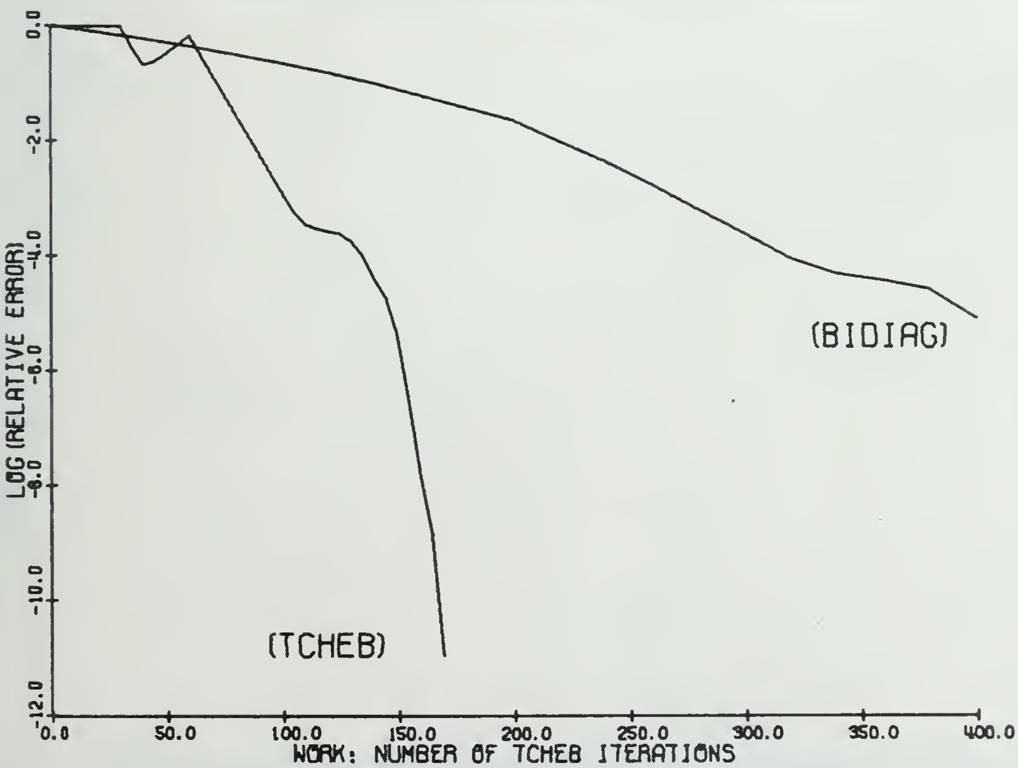

Figure 6.6(b) Work required for $\beta = 4$

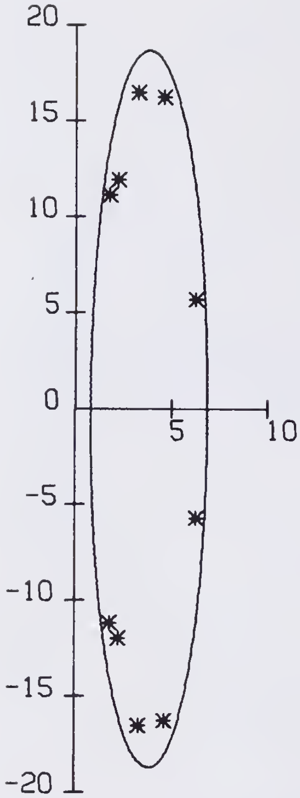

Figure 6.7(a) Best ellipse for β = 8

Figure 6.7(b) Work required for β = 8

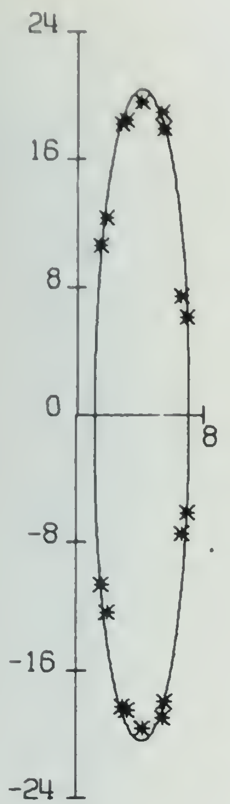

Figure 6.8(a) Best ellipse for β = 10

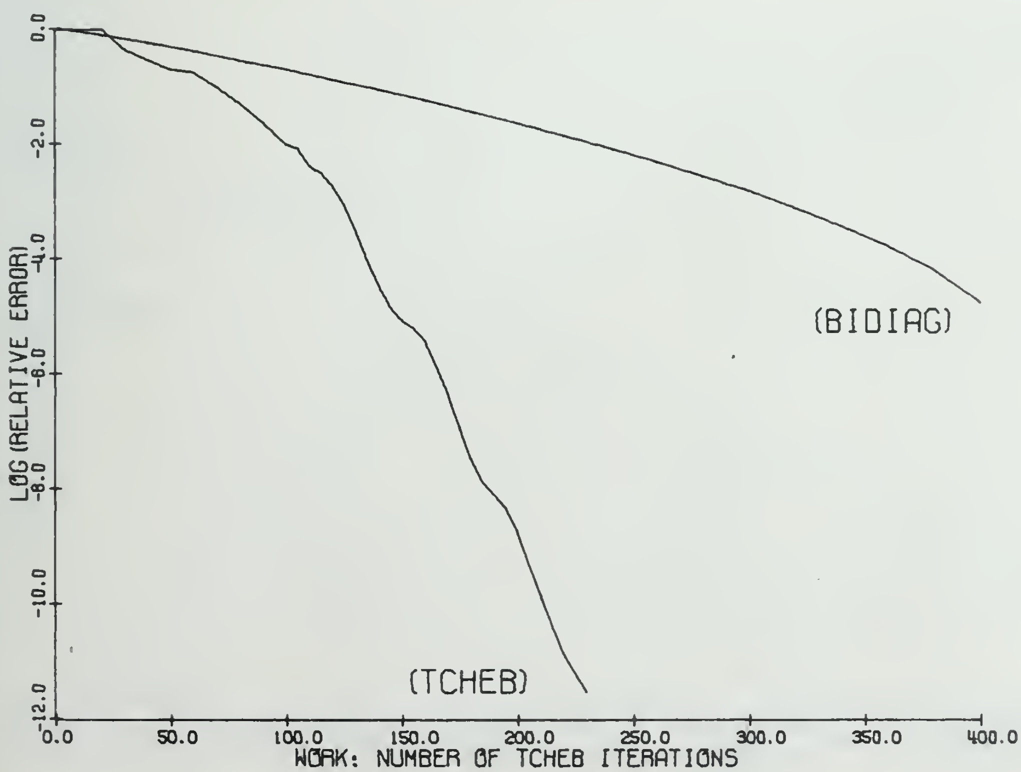

Figure 6.8(b) Work required for β = 10

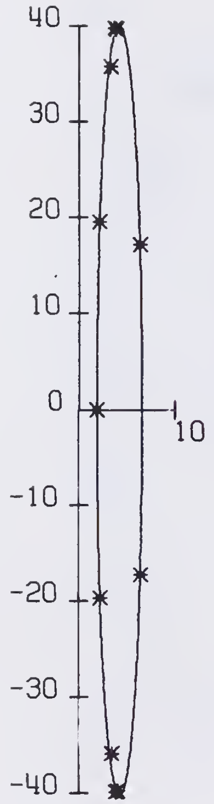

Figure 6.9(a) Best ellipse for β = 20

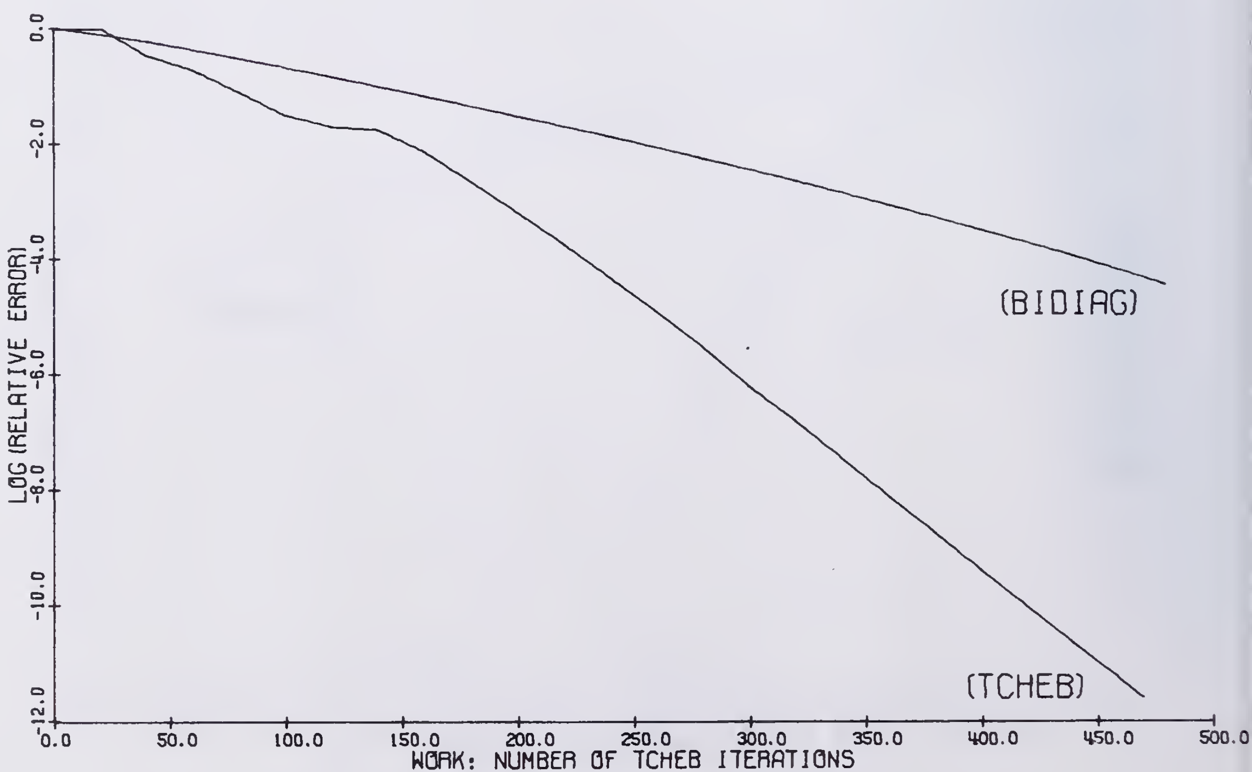

Figure 6.9(b) Work required for β = 20

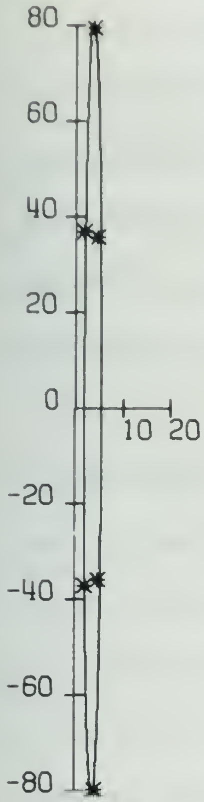

Figure 6.10(a) Best ellipse for β = 40

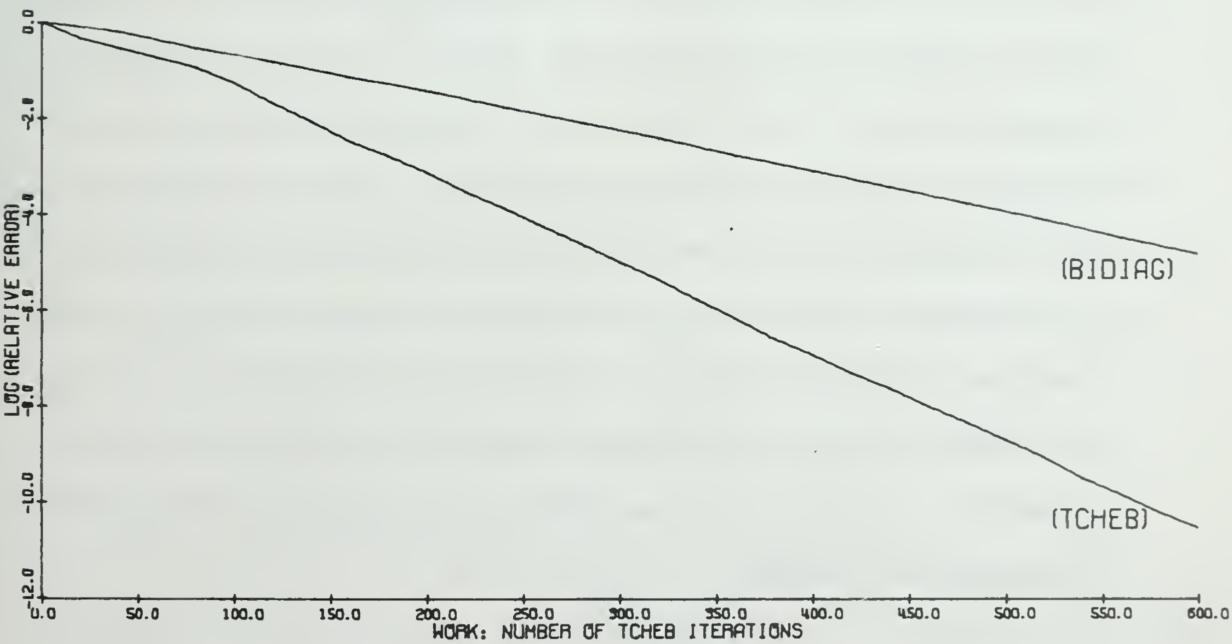

Figure 6.10(b) Work required for β = 40

Figures 6.2(b) - 6.10(b) show a comparison of the error suppression of the two methods, Tchebychef and Bidiagonalization, versus the work required. Error suppression was measured by the log of the relative error; that is, by $\log(\|e_n\|/\|e_0\|)$ where $\|e_n\|$ is the ℓ_2-norm of the error vector at the n^{th} step. Work was measured in terms of the number of Tchebychef iterations. Recall that Bidiagonalization required about wice as much work per iterative step.

The initial choice of parameters d and c2 was based on the rectangle known to contain the spectrum of A. In some cases a rather poor choice was used to show that the adaptive procedure would still work. Table 6.2 contains the initial parameters as well as the computed optimal parameters for each test. (Recall that $c2 = c^2$.) The convergence factor associated with the best ellipse and rate of convergence are also given in Table 6.2. (The rate of convergence is $-\log$ of the convergence factor.)

For A nearly summetric, the condition of A^TA is significantly worse than the condition of A. Since Bidiagonalization is sensitive to the condition of A^TA, it did rather poorly for nearly symmetric A (see Figures 6.2(b) - 6.4(b)). For large β the two methods were more comparable, but the Tchebychef method still held a significant advantage (see Figure 6.10(b)). For these tests the mesh space used was $h = 1$. The symmetric part of A is multiplied by a factor of $1/h^2$, and the antisymmetric part, associated with the first order terms, is multiplied by a factor of $1/h$. For smaller h the matrix would be more nearly symmetric, the type of system for which the Tchebychef algorithm holds the greatest advantage.

In some tests the region of convergence associated with the initial choice of parameters d and c2 did not include the entire spectrum. The error grew in the first cycle. Eigenvalue estimates were extracted, however, and the solution vector was set back to the initial guess. No progress was made toward the solution, but better values for d and c2 were found (see Figures 6.6, 6.8, 6.9). In the test with $\beta = 4$ (see Figure 6.6) two cycles were required before enough information was obtained to choose a d and c2 that produced convergence.

In the tests in which the region of convergence associated with the initial choice of parameters d and c2 contained the entire spectrum, the very rapid initial drop in error was due to the suppression of eigenvalues near the center of the region. The asymptotic rate was achieved after this initial drop (see Figures 6.2, 6.3, 6.4, 6.5, 6.7).

b) Singular Problems

Suppose the matrix A has eigenvalues in the open right half plane except for an eigenvalue at $\lambda = 0$. The Tchebychef algorithm can be applied to this case. Since the scaled and translated Tchebychef polynomials are all normalized at the origin ($P_n(0) = 1$), the component of the initial guess in the direction of the null space of the operator A will not be altered by the Tchebychef algorithm. Indeed, if the parameters d and c2 are chosen to fit the convex hull of nonzero eigenvalues, then a solution will be found containing the component of the null space present in the initial guess. The error in the direction of the nonzero eigenvectors will be suppressed as usual.

A problem would arise, however, if the adaptive procedure were to yield an approximation to the zero eigenvalue. In this case the value

nearly zero would be admitted to the approximate hull and cause a choice of d and c2 that would yield a very slow rate of convergence. However, the eigenvalue estimates are based upon the successive residuals, and if the target vector b is in the range of A, then the residuals are orthogonal to the null space of the matrix A. Since the residuals contain no eigenvectors associated with the zero eigenvalue, the adaptive procedure will not yield an approximation to the zero eigenvalue.

To test the Tchebychef algorithm on a singular system, consider the differential operator above with Neuman boundary conditions. If $Q(x,y) = 0$, then the constant function is a solution of the homogeneous problem

$$\left(-\frac{\partial}{\partial x}\left(A_1(x,y)\frac{\partial}{\partial x}\right) - \frac{\partial}{\partial y}\left(A_2(x,y)\frac{\partial}{\partial y}\right) + B_1(x,y)\frac{\partial}{\partial x} + B_2(x,y)\frac{\partial}{\partial y}\right)u(x,y) = 0 .$$

If we approximate this operator on the same grid as in part a), the matrix A will be the same except that the diagonal must be altered so that the constant vector is in the null space of A. As before let

$$A_1(x,y) = A_2(x,y) = 1,$$

$$B_1(x,y) = B_2(x,y) = \beta .$$

Tests were run with $\beta = 1$ and $\beta = 10$. Figures 6.11(a) and 6.12(a) show the approximate hull of the nonzero eigenvalues and the best ellipse enclosing it. Figures 6.11(b) and 6.12(b) show the error suppression of the two methods, Tchebychef and Bidiagonalization, versus the work required. Error was measured orthogonal to the null space of A.

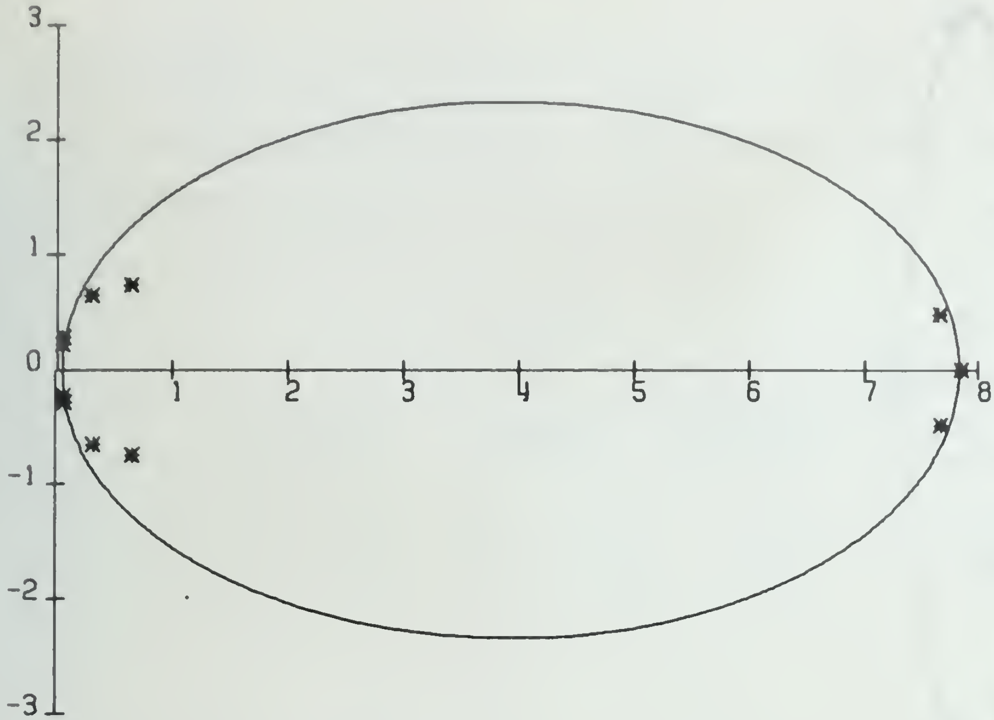

Figure 6.11(a) Best ellipse for β = 1

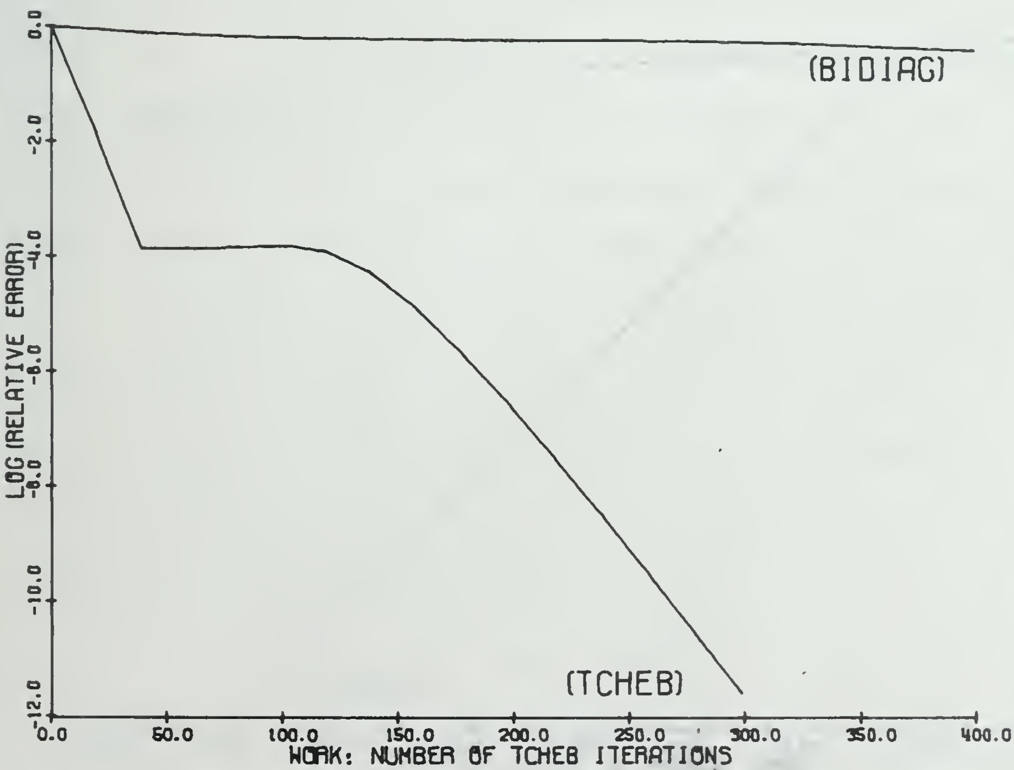

Figure 6.11(b) Work required for β = 1

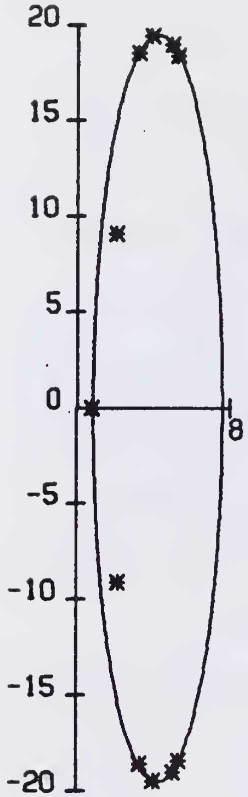

Figure 6.12(a) Best ellipse for β = 10

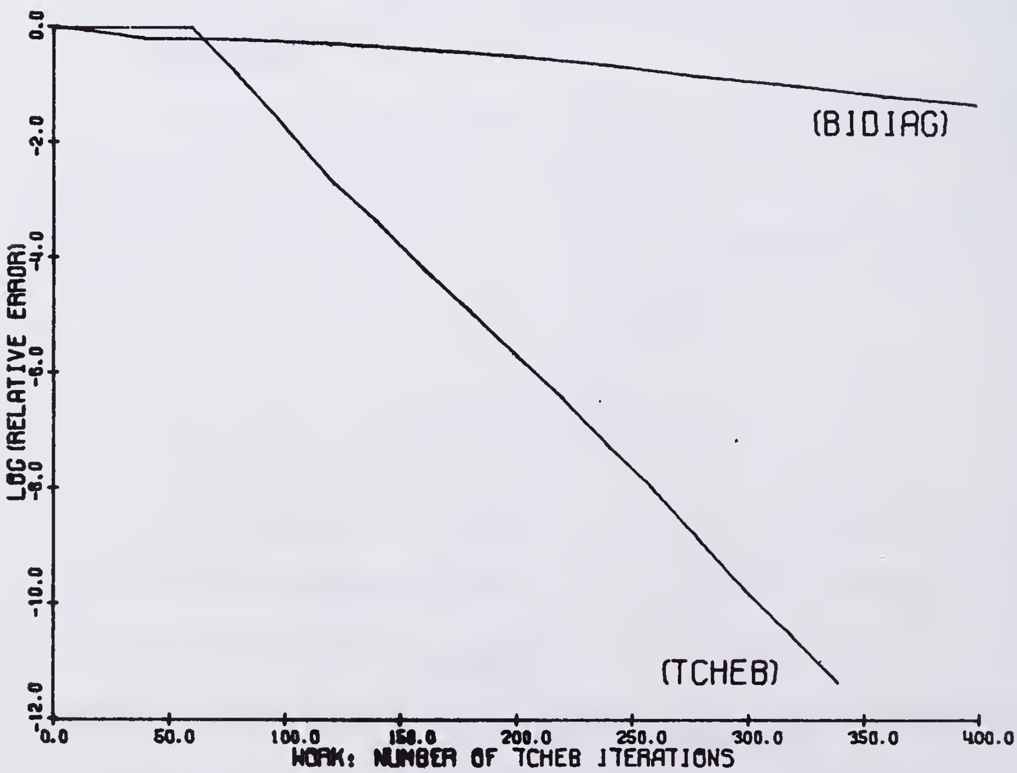

Figure 6.12(b) Work required for β = 10

Starting values for the parameters are listed in Table 6.3 as well as the computed optimal parameters. The convergence factor and rate of convergence also appear in Table 6.3.

Table 6.3

Figure #	β	Initial d	Initial c	Optimal d	Optimal c	Convergence Factor	Rate of Convergence	ICYCLE
11	1	4.0	0	3.93	3.12	.9842	.00691	20
12	10	4.0	16i	4.27	19.11i	.9592	.01809	20

c) Factorization Methods

A recent innovation in iterative methods is factorization (Stone [20]; Dupont, Kendall, and Rachford [5]; Dupont [6]; Kim [14]; Bracha [2]). For the system $Ax = b$, the object is to find a matrix B such that $(A+B)^{-1}$ is easily computable and the equivalent system $(A+B)^{-1}Ax = (A+B)^{-1}b$ has improved condition. Little is known about the spectrum of $(A+B)^{-1}A$, making an adaptive procedure attractive.

One factorization, called the Strongly Implicit Procedure (SIP), is applicable to any 5-point difference matrix (Stone [20]). In this procedure a singular matrix $B(\alpha)$, dependent on the parameter α, is chosen so that $(A+B(\alpha)) = L \cdot U$, where L is lower triangular and U is upper triangular. The equivalent system is

$$(U^{-1}L^{-1}A)x = U^{-1}L^{-1}b \ .$$

When used in conjunction with SIP, the Tchebychef algorithm is computationally unchanged except for computing the residual. Without SIP the residual is given by $r_n = b - Ax_n$. With SIP the residual becomes $r_n = U^{-1}L^{-1}(b - Ax_n)$. This requires 10N multiplications and 10N additions as opposed to 5N. Thus, Tchebychef plus SIP requires 12.7N multiplications and 13.7N additions per step as opposed to 7.7N and 8.7N.

Notice that

$$(A+B(\alpha))^{-1}A = (A^{-1}(A+B(\alpha)))^{-1} = (I + A^{-1}B(\alpha))^{-1} ,$$

so that if μ is an eigenvalue of $A^{-1}B(\alpha)$, then

$$\lambda = \frac{1}{1+\mu}$$

is an eigenvalue of $(A+B(\alpha))^{-1}A$. If the eigenvalues of $A^{-1}B(\alpha)$ are bunched around $\mu = 0$ then the eigenvalues of $(A+B(\alpha))^{-1}A$ will be bunched around $\lambda = 1$. Since $B(\alpha)$ is singular, $\mu = 0$ is an eigenvalue of $A^{-1}B(\alpha)$, so $\lambda = 1$ is an eigenvalue of $(A+B(\alpha))^{-1}A$.

SIP was used in conjunction with the adaptive Tchebychef algorithm on the matrices from part a) with $\beta = .4$ and $\beta = 4$ for several values of α. Figures 6.13(a) and 6.14(a) show the best ellipse enclosing the approximate hull of $(A+B(\alpha))^{-1}A$ for the indicated values of α. Figures 6.13(b) and 6.14(b) show the error suppression versus the number of iterations for the indicated values of α. The iteration parameters d and c2 were initialized to correspond to a circle centered at $\lambda = 1$; that is, d = 1, c2 = 0.

Notice that the eigenvalues of A are transformed into eigenvalues of $(A+B(\alpha))^{-1}A$ that are bunched about $\lambda = 1$. Compare the ellipses

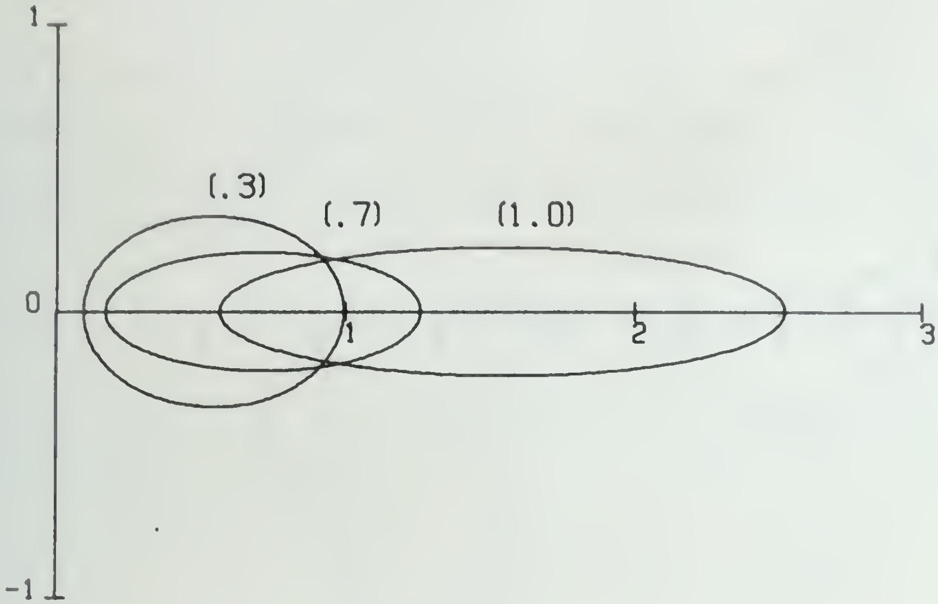

Figure 6.13(a) Best ellipse for indicated value of α and β = .4

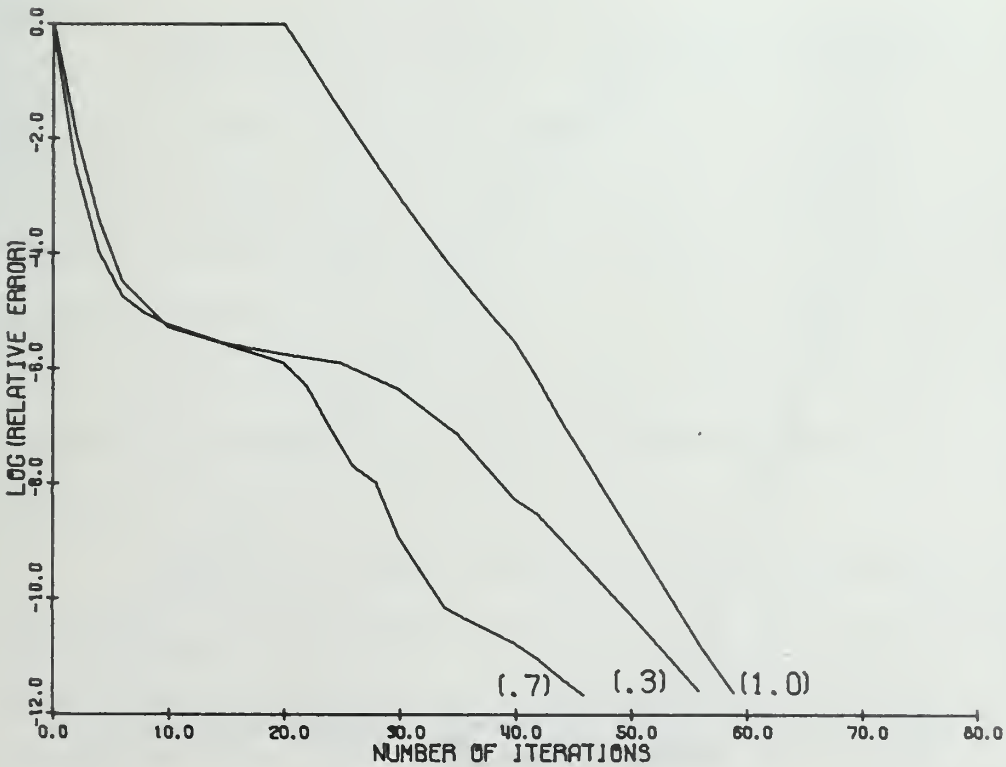

Figure 6.13(b) Work required for indicated value of α and β = .4

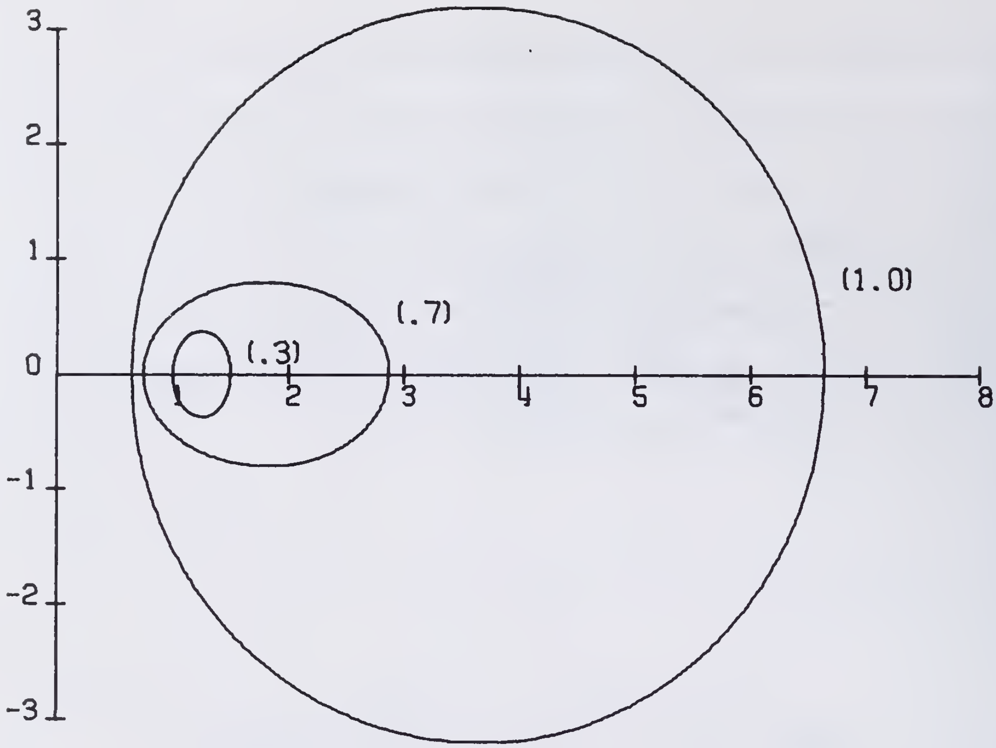

Figure 6.14(a) Best ellipse for indicated value of α and β = 4

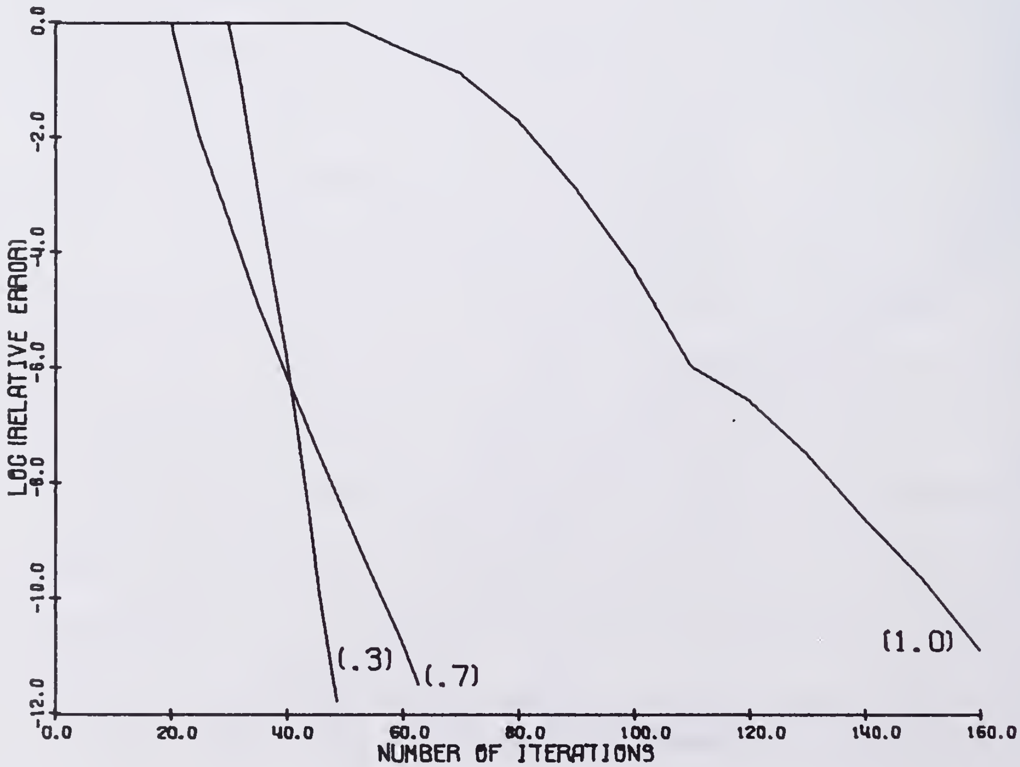

Figure 6.14(b) Work required for indicated value of α and β = 4

of Figures 6.13(a) and 6.14(a) with the ellipses of Figures 6.3(a) and 6.6(a). It is clear from Figures 6.13(b) and 6.14(b) that the conditions of the systems have been greatly improved. Compare the rates of convergence for the same matrices in Table 6.2 and Table 6.4.

Table 6.4

Figure #	β	α	Convergence Factor	Rate of Convergence	ICYCLE
13	.4	.3	.7897	.1025	20
13	.4	.7	.6185	.2086	20
13	.4	1.0	.4371	.3594	20
14	4	.3	.2479	.6057	20
14	4	.7	.5371	.2699	20
14	4	1.0	.8319	.0799	20

Little is known about the effect of the parameter α upon the spectrum of $(A+B(\alpha))^{-1}A$ for nonsymmetric A (for symmetric A see Dupont [5]). It is apparent from the results that the condition of the matrix $(A+B(\alpha))^{-1}A$ is very susceptible to the choice of α. Notice how the ellipses in Figures 6.13(a) and 6.14(a) differ with different α. The accuracy of these ellipses is questionable, however, as convergence occurred too quickly to gain adequate information about the eigenvalue structure.

The greatly improved condition and rapid convergence achieved with SIP more than justifies the extra work per step and presents the explanation of the role of α as an interesting open question.

d) Nonhomogeneous Region

In order to test the algorithm on a more difficult system, the region $0 \leq x \leq 41$, $0 \leq y \leq 41$ was broken up into subregions (see Figure 6.15). The functions $A_1(x,y)$, $A_2(x,y)$, $B_1(x,y)$, $B_2(x,y)$ were given values as follows:

Region A

$A_1(x,y) = 10$

$A_2(x,y) = 10$

$B_1(x,y) = 0$

$B_2(x,y) = 0$

Region B

$A_1(x,y) = 10$

$A_2(x,y) = 10$

$B_1(x,y) = 10$

$B_2(x,y) = 10$

Region C

$A_1(x,y) = 100$

$A_2(x,y) = 1$

$B_1(x,y) = 10$

$B_2(x,y) = 0$

Region D

$A_1(x,y) = 1$

$A_2(x,y) = 100$

$B_1(x,y) = 0$

$B_2(x,y) = 10$

A grid with mesh space $h = 1$ and Dirichlet boundary conditions were used to generate the associated 5-point difference operator, A, of dimension 1600. The adaptive Tchebychef algorithm was used on this system, both by itself and with SIP. The results are shown in Figure 6.16. Figure 6.16(b) shows the approximate hull of the matrix A. Figure 6.16(a) shows the approximate hull of the matrix $(A+B(\alpha))^{-1}A$ with $\alpha = .5$. Figure 6.16(c) shows the error suppression of the Tchebychef

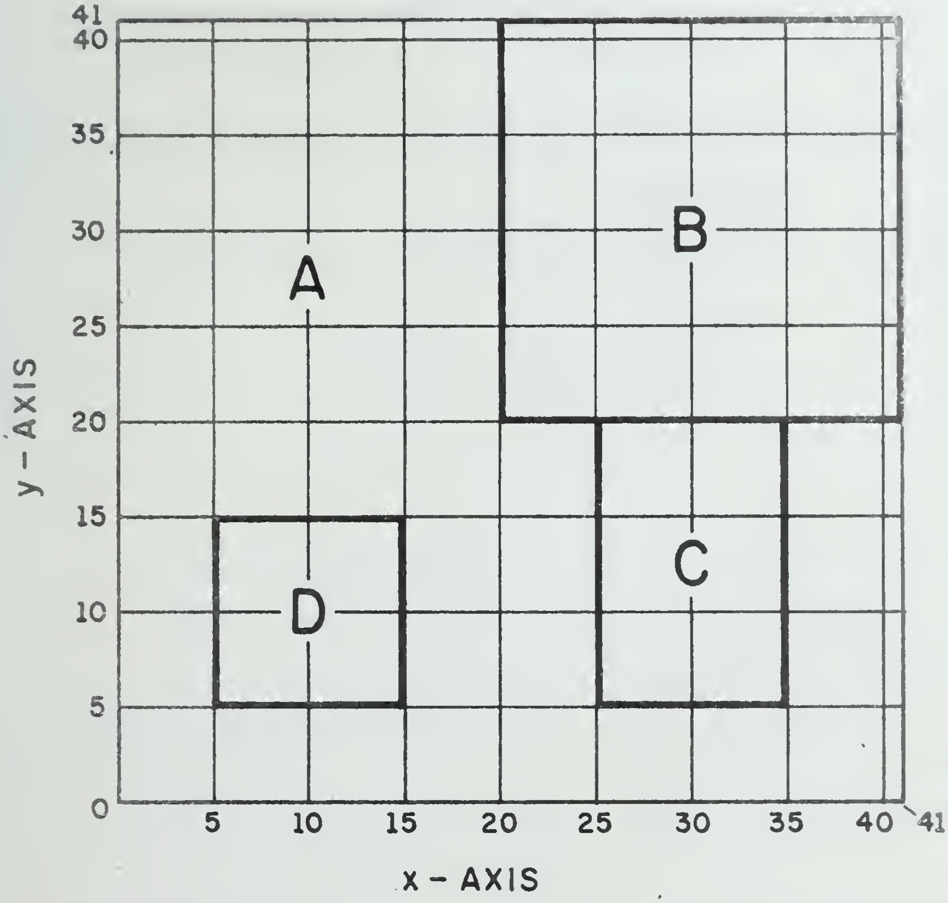

Figure 6.15

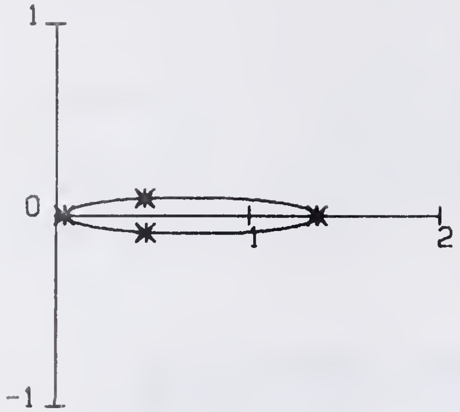

Figure 6.16(a) Spectrum of $(A+B(\alpha))^{-1}A$

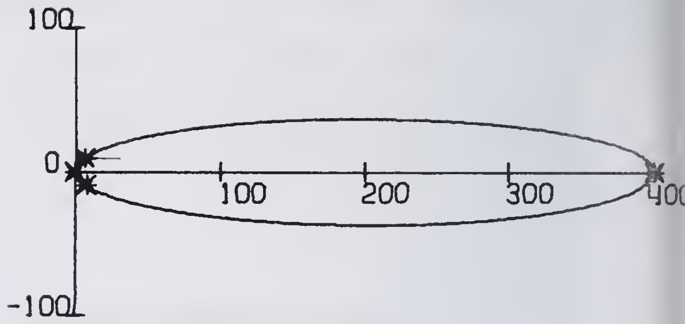

Figure 6.16(b) Spectrum of A

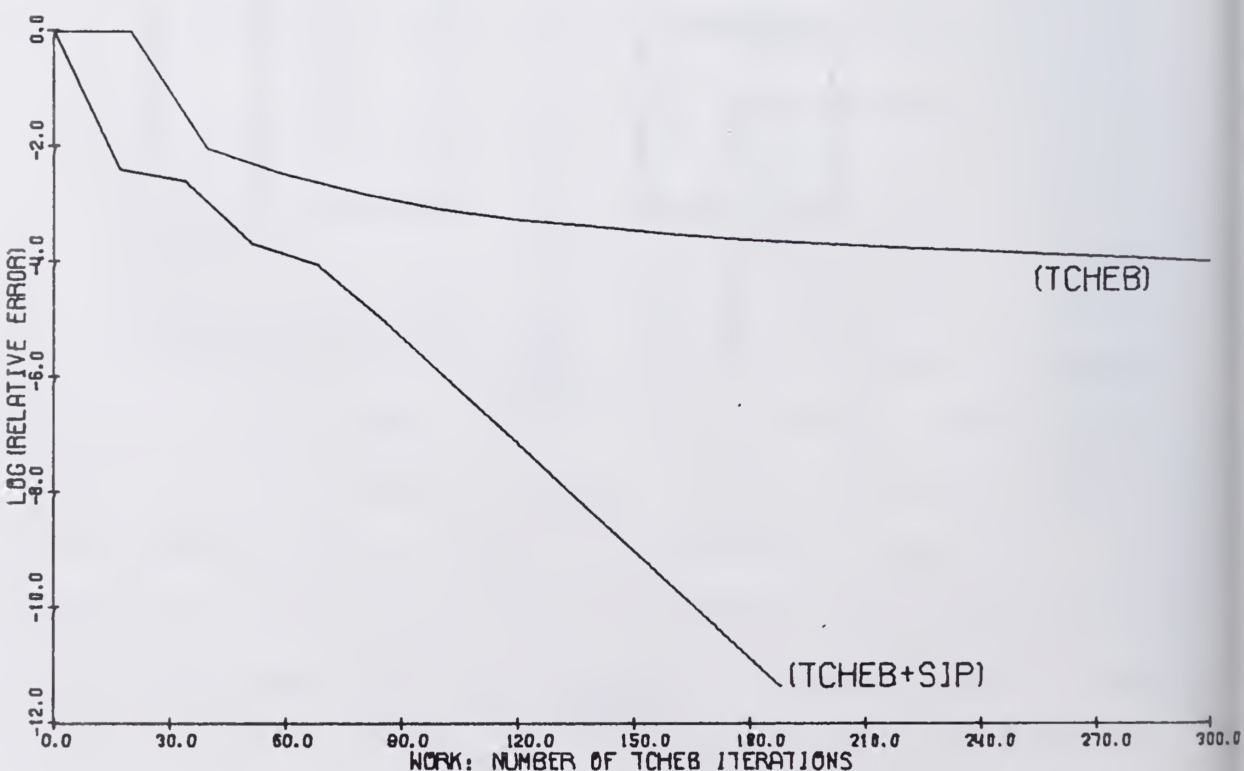

Figure 6.16(c) Work required

method, alone and with SIP, versus work. The work is measured in the number of Tchebychef iterations; the number of iterations with SIP is multiplied by a factor of 12/7.

Notice that the entire spectrum of A is transformed into a spectrum closely bunched about $\lambda = 1$. The condition of the system is greatly improved. The convergence factors and rates of convergence of the two systems are given in Table 6.5.

Table 6.5

	Initial		Optimal		Convergence Factor	Rate of Convergence	ICYCLE
	d	c	d	c			
TCHEB	40	0	201.61	198.11	.9984	.00070	20
TCHEB + SIP	1	0	.7072	.6509	.7840	.1056	20

In a problem such as this, as much is known about the spectrum of $(A+B(\alpha))^{-1}A$ as is known about the spectrum of the matrix A. The initial choice of parameters, as well as the computed optimal choice of parameters, is given in Table 6.5. In both cases the adaptive procedure was able to recover from poor initial parameters and produce good estimates of the optimal parameters.

The choice of $\alpha = .5$ was quite arbitrary. With the value $\alpha = 1.0$, the matrix $(A+B(\alpha))^{-1}A$ had an eigenvalue with negative real part. The improvement in the condition of the system with $\alpha = .5$ is significant enough to warrant further study into the role of α in this factorization.

6.4 Summary

The adaptive Tchebchef algorithm has qualities that promote its use on nonsymmetric systems whose eigenvalues lie in the right half plane. The method does not depend upon any special structure of the matrix. It can be used on finite element matrices and difference matrices, as well as in conjunction with factorization methods.

The Tchebychef algorithm requires less than half of the work per iterative step that is required by the two competing methods: Bidiagonalization and Conjugate Gradients on A^TA. The additional storage required by the adaptive procedure of the Tchebychef algorithm becomes less of a factor as the number of diagonals needed to store the matrix A increases.

In the tests that were run, the Tchebychef algorithm was considerably faster than the competing methods. This is due, in part, to the fact that it is sensitive to the condition of A, whereas the competing methods are sensitive to the condition of A^TA. This advantage was especially apparent on nearly symmetric systems.

Another quality of the Tchebychef method is stability. The Tchebychef algorithm makes steady, predictable progress toward the solution. At each step the error is multiplied by a factor that is no greater than the convergence factor. This factor is determined by the choice of parameters d and c2 and the spectrum of the matrix. Any round-off error will be multiplied by this factor on the next step. The competing methods, on the other hand, depend upon orthogonality and A-orthogonality (A^TA-orthogonality). The rate of error reduction of these methods is unpredictable at best, and error reduction may become very slow if orthogonality breaks down (Hestenes and Stiefel [12]).

The adaptive procedure was shown to be able to produce good estimates of the optimal iteration parameters with virtually no a priori knowledge of the spectrum of the matrix A. Although the stability of the adaptive procedure could not be guaranteed (Section 5.1), in all of the tests the procedure behaved as expected. All of the eigenvalue estimates lay inside the true hull of the eigenvalues, and, after good parameters were found, further eigenvalue estimates lay near the center of the approximate hull as predicted. This may be due in part to the absence of nonlinear elementary divisors of large dimension in the test matrices. Further refinement of the adaptive procedure may be necessary in the presence of nonlinear elementary divisors of large dimension.

In addition to finding good iteration parameters, the adaptive procedure provides information about the spectrum of the matrix that may be useful to the user. One such use is in the choice of the error bound (Section 6.1).

The adaptive Tchebychef algorithm showed great promise when used in conjunction with SIP. Although the properties of the parameter α are not understood, the significant improvement in the condition of the systems tested showed the potential of this factorization method.

The Tchebychef algorithm could be used in conjunction with other factorization methods as well. Y. J. Kim [14] has shown that some finite element matrices can be factored in much the same way as the 5-point difference operators are factored by SIP. It has also been suggested that the Tchebychef algorithm could be used in conjunction

with Symmetric Successive Over Relaxation. In each of these factorizations, the adaptive property makes the Tchebychef method attractive because little is known about the spectrum of the equivalent system.

LIST OF REFERENCES

[1] Birkhoff, G. and S. Maclane, A Survey of Modern Algebra, MacMillan, New York, 1953.

[2] Bracha, A., "A Symmetric Factorization Procedure for the Solution of Elliptic Boundary Value Problems," Digital Computer Laboratory Reports, Rep. No. 440, University of Ill., April 1971.

[3] Diamond, M. A., "An Economical Algorithm for the Solution of Elliptic Difference Equations Independent of User-Supplied Parameters," Ph.D. Dissertation, Department of Computer Science, University of Ill., 1972.

[4] Dunford, N. and J. L. Schwartz, Linear Operators, Interscience Publishers, New York, 1958.

[5] Dupont, T., R. R. Kendall, and H. H. Rachford Jr., "An Approximate Factorization Procedure for Solving Self-Adjoint Elliptic Difference Equations," SIAM J. Numer. Anal., Vol. 5, Sept. 1968, p. 558.

[6] Dupont, T., "A Factorization Procedure for the Solution of Elliptic Difference Equations," SIAM J. Numer. Anal., Vol. 5, Dec. 1968, p. 753.

[7] Engeli, M., T. H. Ginsburg, H. Rutishauser, and E. L. Stiefel, "Refined Iterative Methods for Computation of the Solution and Eigenvalues of Self-Adjoint Boundary Value Problems," Mitteilungen aus dem Institute fur Angewandte Mathematik, No. 8, 1959.

[8] Faddeev, D. K. and U. N. Faddeeva, Computational Methods of Linear Algebra, W. H. Freeman & Co., San Francisco, 1963.

[9] Fox, L. and I. B. Parker, Chebyshev Polynomials in Numerical Analysis, London, Oxford University Press, 1968.

[10] Golub, G. H. and R. S. Varga, "Chebyshev Semi-iterative Methods, Successive Over Relaxation Iterative Methods and Second Order Richardson Iterative Methods," Numerische Mathematik, Vol. 3, 1961, p. 147.

[11] Golub, G. and W. Kahan, "Calculating the Singular Values and Pseudo-Inverse of a Matrix," SIAM J. Numer. Anal., Vol. 2, 1965.

[12] Hestenes, M. R. and E. L. Stiefel, "Methods of Conjugate Gradients for Solving Linear Systems," N.B.S. J. of Res., Vol. 49, 1952, p. 409.

[13] Hille, E., Analytic Function Theory, Vol. II, Ginn & Co., Boston, 1962, Ch. 16, pp. 264-274.

[14] Kim, Y. J., "An Efficient Iterative Procedure for Use with the Finite Element Method," Ph.D. Dissertation, Department of Computer Science, University of Ill., UIUCDCS-R-73-600, August 1973.

[15] Kincaid, D. R., "On Complex Second-Degree Iterative Methods," Center for Numerical Analysis, University of Texas, Austin, 1968.

[16] Kjellberg, G., "On the Convergence of Successive Over Relaxation Applied to a Class of Linear Systems of Equations with Complex Eigenvalues," Ericsson Technics, No. 2, 1958.

[17] Paige, C. C., "Bidiagonalization of Matrices and Solution of Linear Equations," SIAM J. Numer Anal., Vol. 11, March 1974, p. 197.

[18] Shampine, L. F. and R. C. Allen Jr., Numerical Computing: An Introduction, W. B. Saunders Co., Philadelphia, 1973.

[19] Stiefel, E. L., "Kernel Polynomials in Linear Algebra and Their Applications," U.S. N.B.S. Applied Math Series, Vol. 49, Jan. 1958, p. 1.

[20] Stone, H. L., "Iterative Solutions of Implicit Approximations of Multidimensional Partial Differential Equations," SIAM J. Numer. Anal., Vol. 5, Sept. 1968, p. 530.

[21] Varga, R. S., "A Comparison of Successive Over Relaxation and Semi-Iterative Methods Using Chebyshev Polynomials," SIAM J. Numer Anal., Vol. 5, 1957, p. 39.

[22] Varga, R. S., Matrix Iterative Analysis, Prentice-Hall, Englewood Cliffs, New Jersey, 1962.

[23] Walsh, J. L., Interpolation and Approximation by Rational Functions in the Complex Domain, Revised Edition, Colloquium Publications, Vol. 20, A.M.S., Providence, R.I., 1956.

[24] Wilkinson, J. H., The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965.

[25] Wrigley, H. E., "Accelerating the Jacobi Method for Solving Simultaneous Equations by Chebyshev Extrapolation when the Eigenvalues of the Iteration Matrix are Complex," Computer Journal, Vol. 6, July 1963, p. 169.

[26] Young, D., Iterative Solution of Large Linear Systems, Academic Press, New York and London, 1971

[27] Zienkiewicz, O. C., The Finite Element Method in Engineering Science, McGraw-Hill, London, New York, 1971.

APPENDIX A

```
C     SUBROUTINE TCHEB(A,X,B,R,DX,XLAST,RLAST,S,CH,
C    1                            ERBND,ICYCLE,IMAX)
C ****************************************************************** *
C     THIS SUBROUTINE SOLVES THE SYSTEM AX=B, RETURNING THE          *
C SOLUTION, X. THE POSITIVE HULL OF THE EIGENVALUES OF A IS          *
C DETERMINED DYNAMICALLY AND RETURNED IN CH.                         *
C--------------------------------------------------------------------*
C                                                                    *
C    THE USER MUST SUPPLY THE SUBROUTINE NSYMAX(Y,X,A),              *
C    WHICH PREFORMS THE MATRIX VECTOR MULTIPLLICATION:               *
C                   Y = AX.                                          *
C                                                                    *
C    THE INPUT PARAMETERS ARE:                                       *
C                                                                    *
C       A(N,NC)           THE MATRIX                                 *
C       X(N)              THE INITIAL GEUSS                          *
C       B(N)              THE TARGET VECTOR                          *
C                                                                    *
C       ERBND             THE ACCEPTABLE ERROR                       *
C       ICYCLE            THE NUMBER OF ITERATIVE STEPS BETWEEN      *
C                         ADAPTIVE PROCEDURES                        *
C       ITMAX             THE MAXIMUM NUMBER OF ITERATINS ALLOWED    *
C                                                                    *
C       COMMON /AREA1/ N,NC :                                        *
C               N         THE DIMENSION                              *
C               NC        THE NUMBER OF COLUMNS NEEDED TO STORE      *
C                         THE MATRIX A(N,NC)                         *
C                                                                    *
C       COMMON /AREA2/ D,C2,A2,FC :                                  *
C               D         THE CENTER OF THE ELLIPSE                  *
C               C2        THE FOCAL LENGTH SQUARED                   *
C                                                                    *
C--------------------------------------------------------------------*
C                                                                    *
C       OTHER VARIABLES ARE:                                         *
C                                                                    *
C               R(N)              THE RESIDUAL                       *
C               DX(N)             THE CHANGE IN X AT EACH STEP       *
C               XLAST(N)          X AT THE START OF THE CYCLE        *
C               RLAST(N)          R AT THE START OF THE CYCLE        *
C               S(N,4)            THE LAST 4 RESIDUALS OF            *
C                                 EACH CYCLE                         *
C                                                                    *
C               CH(25,2)          EIGENVALUE ESTIMATES               *
C               ICH(25,2)         LINK LISTING OF CH                 *
C                                                                    *
C               A2                REAL AXIS OF THE BEST ELLIPSE      *
C               CF                CONVERGENCE FACTOR OF THE          *
```

```
C                                    BEST ELLIPSE                        *
C               RSD                  RESIDU: NORM OF RESIDUAL            *
C               RO                   RESIDU AT THE START OF              *
C                                    THE CYCLE                           *
C               P1                   ITERATION PARAMETERS                *
C               P2                                                       *
C                                                                        *
C ********************************************************************** *
      SUBROUTINE TCHEB(A,X,B,R,DX,XLAST,RLAST,S,CH,
     1                           ERBND,ICYCLE,IMAX)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /AREA1/ N,ND
      COMMON /AREA2/ D,C2,A2,FC
      COMMON /AREA3/ ICH,IFIRST,IFREE
      DIMENSION A(N,ND),X(N),B(N),R(N),DX(N),
     1               XLAST(N),RLAST(N),S(N,4)
      DIMENSION CH(25,2),ICH(25,2)
C
C     BEGIN
C
C     INITIALIZE CH,ICH,IFIRST,IFREE,A2,FC
         CALL INIT(CH)
C
C     INITIALIZE ISTEP
          ISTEP = 0
C
C     CALCULATE THE INITIAL RESIDUAL
    6     CALL NSYMAX(R,X,A)
          DO 8 I=1,N
             R(I) = B(I) - R(I)
    8     CONTINUE
C
C     OUTPUT THE INITIAL  RESIDU
          CALL OUTPUT(R,RSD,ISTEP)
C
C     SAVE INITIAL X AND R
          RC = RSD
          CALL LASTX(X,R,RSD,XLAST,RLAST,RC)
C
C     BEGIN STIEFEL ITERATION
C
C         INITIALIZE DX AND PARAMETERS P1 AND P2
   10             DO 15 I = 1,N
                     DX(I) = R(I)/D
   15             CONTINUE
C
                  P1 = 2.D0/D
                  P2 = 0.D0
```

```
C
C       MAIN LOOP
   20          DO 40 I = 1,ICYCLE
C
C                     STORE LAST FOUR RESIDUALS IN S
                           IF(I.GT.ICYCLE-4) GO TO 25
                                GO TO 30
C                          THEN STORE R
   25                           ICOL=ICYCLE-I+1
                                DO 26 K=1,N
                                     S(K,ICOL)=R(K)
   26                           CONTINUE
C
C                     COMPUTE NEW X
   30                      DO 32 J=1,N
                                X(J) = X(J) + DX(J)
   32                      CONTINUE
C
C                     COMPUTE NEW R
                           CALL NSYMAX(R,X,A)
                           DO 34 J=1,N
                                R(J) = B(J) - R(J)
   34                      CONTINUE
C
C                     COMPUTE NEW PARAMETERS
                           P2 = C2*P1/(4.D0*D - C2*P1)
                           P1 = (1.D0 + P2)/D
C
C                     COMPUTE NEW DX
                           DO 36 J=1,N
                                DX(J) = P1*R(J) + P2*DX(J)
   36                      CONTINUE
C
C                     UPDATE ISTEP
                           ISTEP = ISTEP+1
C
C                     OUTPUT RESIDU
                           CALL OUTPUT(R,RSD,ISTEP)
C
   40          CONTINUE
C
C       END MAIN LOOP
C
C    END STIEFEL
C
C    TEST TO HALT
        IF(RSD.LT.ERBND) GO TO 90
        IF(ISTEP.GE.IMAX) GO TO 90
```

```
C
C      BEGIN ADAPTIVE PROCEDURE
             CALL ADAPT(S,R,CR,IFLAG)
C
C            IF D OR C2 HAVE BEEN CHANGED RESTART STIEFEL,
C            OTHERWISE RESUME STIEFEL
                    IF(IFLAG.EQ.1) GO TO 45
                           GO TO 50
C                   THEN RESTART STIEFEL
   45                      CALL LASTX(X,R,RSD,XLAST,RLAST,FC)
                           GO TO 10
C                   ELSE RESUME STIEFEL
   50                      GO TO 20
C
C      END ADAPTIVE PROCEDURE
C
   90 RETURN
      END




      SUBROUTINE OUTPUT(R,RSD,ISTEP)
C *****************************************************************
C   THIS SUBROUTINE OUTPUTS THE ITERATION PARAMETERS D AND C2,    *
C   THE EXPECTED CONVERGENCE FACTOR, FC, AND THE NORM OF R.        *
C *****************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /AREA1/ N,ND
      COMMON /AREA2/ D,C2,A2,FC
      DIMENSION R(N)
C
C     BEGIN
          CALL INPRO(R,R,RSD,N)
          RSD = DSQRT(RSD)
C
          WRITE(6,100) ISTEP,D,C2,FC,RSD
  100     FORMAT(' ISTEP =',I3,3X,'D = ',D12.5,3X,'C2 = ',
     1               D12.5,3X,'FC = ',D12.5,3X,'RESIDU = ',D12.5)
C     END
C
      RETURN
      END
```

```
      SUBROUTINE LASTX(X,R,RSD,XLAST,RLAST,RO)
C*******************************************************************
C   THIS SUBROUTINE  REPLACES X AND R BY PREVIOUS VALUES          *
C   OF X AND R IF THE RESIDU, RSD, HAS GROWN.                     *
C ******************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /AREA1/ N,ND
      DIMENSION X(N),R(N),XLAST(N),RLAST(N)
C
C     BEGIN
          IF(RSD.GT.RO) GO TO 10
               GO TO 20
C         THEN CHANGE X AND R
   10          DO 15 I=1,N
                    X(I)=XLAST(I)
                    R(I)=RLAST(I)
   15          CONTINUE
               RSD=RO
               GO TO 30
C         ELSE UPDATE XLAST AND RLAST
   20          RO = RSD
               DO 25 I=1,N
                    XLAST(I)=X(I)
                    RLAST(I)=R(I)
   25          CONTINUE
C     END
C
   30 RETURN
      END




      SUBROUTINE INPRD(X,Y,PROD,N)
C ******************************************************************
C     THIS SUBROUTINE FINDS THE INNER PRODUCT OF THE TWO          *
C VECTORS: PROD = <X(N),Y(N)>                                     *
C ******************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(N),Y(N)
C
C     BEGIN
          PROD = 0.D0
          DO 10 I=1,N
               PROD = PROD + X(I)*Y(I)
   10     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INIT(CH)
C*****************************************************************
C     THIS SUBROUTINE INITIALIZES CH,ICH,IFIRST,IFREE,A2,FC *
C THE HULL IS GIVEN THE VALUES D+C, D-C AND THE ELLIPSE IS *
C THE LINE BETWEEN THEM.                                    *
C*****************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /AREA2/ D,C2,A2,FC
      COMMON /AREA3/ ICH,IFIRST,IFREE
      DIMENSION CH(25,2),ICH(25,2)
C
C     BEGIN
C
C     INITIALIZE ICH
         DO 5 I=1,25
            ICH(I,1) = I-1
    5       ICH(I,2) = I+1
C
C     INITIALIZE CH,IFIRST,IFREE,A2,FC
         IFIRST = 1
         IF(C2.LE.0.0D0) GO TO 10
            GO TO 20
C        THEN
   10       IFREE = 2
            ICH(1,1) = 1
            ICH(1,2) = 1
            CH(1,1) = D
            CH(1,2) = DSQRT(-C2)
            A2 = 0.0D0
            GO TO 30
C        ELSE (C2.GT.0)
   20       IFREE = 3
            ICH(1,1) = 1
            ICH(1,2) = 2
            ICH(2,1) = 1
            ICH(2,2) = 2
            CH(1,1) = D-DSQRT(C2)
            CH(1,2) = 0.0D0
            CH(2,1) = D+DSQRT(C2)
            CH(2,2) = 0.0D0
            A2 = C2
C
   30 FC = 0.0D0
C
      RETURN
      END
```

```
      SUBROUTINE ADAPT(S,R,CH,IFLAG)
C**********************************************************************
C     THIS SUBROUTINE FINDS THE BEST ITERATION PARAMETERS            *
C BASED ON THE CONVEX HULL OF A SET OF APPROXIMATE EIGEN-            *
C VALUES, GENERATED FROM THE RESIDUAL VECTORS STORED IN THE          *
C MATREX S.                                                          *
C**********************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /AREA1/ N,ND
      COMMON /AREA2/ D,C2,A2,FC
      DIMENSION S(N,4),R(N)
      DIMENSION CH(25,2),Q(4),EV(4,2)
C
C     FIND THE LEAST SQUARE SOLUTION TO  SQ+R=0
          CALL LSTSQ(S,Q,R)
C
C     FIND THE EIGENVALUE APPROXIMATIONS
          CALL EVS(Q,EV,IRT)
C
C     IF NO NEW EIGENVALUES RESUME STIEFEL ITERATION
          IF(IRT.EQ.0) GO TO 5
              GO TO 10
C         THEN
    5         IFLAG = 0
              GO TO 20
C
C     ADD THE NEW APPROXIMATIONS TO THE PREVIOUS CONVEX HULL
C     AND FORM THE NEW CONVEX HULL
   10     CALL HULL(CH,EV,IFLAG,IRT)
C
C     IF THE EIGENVALUE APPROXIMATIONS ADD NO NEW INFORMATION
C     THEN RESUME STIEFEL ITERATION
          IF(IFLAG.EQ.0) GO TO 20
C
C     FIND THE BEST PARAMETERS FOR THE HULL
          CALL ELLIP(CH)
C
   20 RETURN
      END
```

```
      SUBROUTINE LSTSQ(S,Q,R)
C**********************************************************************
C     THIS SUBROUTINE FINDS THE LEAST SQUARES SOLUTION OF            *
C THE SYSTEM, S*Q = -R, BY SOLVING THE SYSTEM, STS*Q = B,            *
C WHERE ST = S-TRANSPOSE AND B =-ST*R. A BIDIAGONALIZATION *
C ROUTINE IS USED TO SOLVE THE SMALL SYSTEM.                         *
C**********************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /AREA1/ N,ND
      DIMENSION S(N,4),Q(4),R(N),STS(4,4),B(4)
C
C     COMPUTE ST*S
         DO 10 I=1,4
            DO 10 J=I,4
               CALL INPRO(S(1,I),S(1,J),STS(I,J),N)
               STS(J,I) = STS(I,J)
   10    CONTINUE
C
C     COMPUTE B
         DO 20 I=1,4
            CALL INPRO(S(1,I),R,B(I),N)
            B(I) =-B(I)
   20    CONTINUE
C
C     NORMALIZE THE SYSTEM
         ALPHA = STS(1,1)
         DO 30 I=1,4
            B(I) = B(I)/ALPHA
            DO 30 J=1,4
               STS(I,J) = STS(I,J)/ALPHA
   30    CONTINUE
C
C     SOLVE THE SYSTEM STS*Q = B
C  ***ANY LIBRARY ROUTINE MAY BE USED TO SOLVE ***
C  ***      THE 4X4 SYSTEM :  STS*Q = B.        ***
C  ***      BIDIAG APPEARS IN APPENDIX B.       ***
C
            CALL BIDIAG(STS,Q,B)
C
      RETURN
      END
```

```
      SUBROUTINE EVS(Q,EV,IRT)
C*****************************************************************
C     THIS SUBRCUTINE FINDS THE RCCTS OF THE PCLYNOMIAL          *
C WITH THE COEFICIENTS: 1,Q(1),Q(2),C(3),C(4). THESE RCCTS       *
C ARE THEN TRANSFORMEC INTO EIGENVALUES OF THE MATREX A.         *
C*****************************************************************
      IMPLICTT REAL*8(A-H,O-Z)
      CCMMCN /AREA2/ C,C2,A2,FC
      DIMENSICN AA(51,3),RCCTS(2,50),CD(51)
      DIMENSICN EV(4,2),Q(4)
C
C     BEGIN
C
C     FINC RCCTS
C
C  ***ANY LIBRARY ROUTINE MAY BE USED TO FIND THE RCCTS.       ***
C  ***THE RCCTS SHOULD BE STOREC IN EV(4,2) FROM THE FRONT.    ***
C  ***EV(I,1) IS THE REAL PART AND EV(I,2) THE IMAGINARY PART.***
C  ***THE VARIABLE IRT INDICATES THE NUMER CF RCCTS FCUND.     ***
C  ***          RSSR APPEARS IN APPENDIX C.                    ***
C
          AA(1,1) = 1.0D0
          DC 10 I=1,4
             AA(I+1,1) = Q(I)
  10      CONTINUE
          IDEG = 4
C
          CALL RSSR(AA,ROOTS,IDEG,10,15,1.0D-4,1.0D-6,DD)
C
          IRT = 4-IDEG
          DO 14 I=1,IRT
             EV(I,1) = RCCTS(1,5-I)
             EV(I,2) = ROCTS(2,5-I)
  14      CONTINUE
C
C
C     TRANSFORM THE ROOTS TO EIGENVALUES CF A
          G = D+DSQRT(D*D-C2)
          K=IRT
          II=0
          DC 20 I=1,K
             T1=EV(I,1)*EV(I,1)+EV(I,2)*EV(I,2)
             IF(T1.LT.CABS(C2)/(G*G)) GO TC 16
                GC TC 18
C            THEN DISCARD RCCT
  16            IRT=IRT-1
                GC TC 20
C            ELSE FIND EIGENVALUE ESTIMATE
```

```
   18               II=II+1
                    EV(II,1)=D-.5D0*EV(I,1)*(C2/(T1*G)+G)
                    EV(II,2)=-.5D0*EV(I,2)*(C2/(T1*G)-G)
   20         CONTINUE
C
C       OUTPUT NEW EIGENVALUE ESTIMATES
              IF(IRT.EQ.0) GO TO 22
                 GO TO 24
C             THEN
   22            WRITE(6,300)
                 GO TO 40
C             ELSE
   24            WRITE(6,100)
                 DO 30 I=1,IRT
                    WRITE(6,200) EV(I,1),EV(I,2)
   30            CONTINUE
  100         FORMAT('0THE NEW EIGENVALUE ESTIMATES ARE:')
  200         FORMAT(' ',5X,D12.5,' + I* ',D12.5)
  300         FORMAT('0NO NEW EIGENVALUE ESTIMATES')
   40 RETURN
      END
```

```
      SUBROUTINE HULL(CH,EV,IFLAG,IRT)
C*********************************************************************
C     THIS SUBROUTINE FORMS THE POSITIVE HULL OF THE UNION  OF *
C THE PREVIOUS POSITIVE HULL AND THE NEW EIGENVALUE ESTIMATES. *
C IF NONE OF THE NEW EIGENVALUE ESTIMATES ARE IN THE POSITIVE  *
C HULL, IFLAG IS RETURNED AS 0.                                *
C*********************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /AREA2/ D,C2,A2,FC
      COMMON /AREA3/ ICH,IFIRST,IFREE
      DIMENSION CH(25,2),ICH(25,2),EV(4,2)
C
C     BEGIN
C
C
C     SET IFLAG
          IFLAG = 0
C
C     PLACE NEW EIGENVALUE ESTIMATES IN THE HULL
          IF(IRT.EQ.0) GO TO 90
          DO 50 I=1,IRT
              IF(EV(I,2).LT.0.0D0) GO TO 50
C
              TEST=(D-EV(I,1))*(D-EV(I,1))*(A2-C2)+
     1            EV(I,2)*EV(I,2)*A2-A2*(A2-C2)
              IF(TEST.LT.0.0D0) GO TO 2
                  GO TO 4
C             THEN EV(I,-) IS NOT IN THEN HULL
    2             GO TO 50
C             ELSE SET IFLAG
    4             IFLAG = 1
C
C             PUT EV IN NEXT OPEN SPOT
                  J=IFREE
                  CH(J,1)=EV(I,1)
                  CH(J,2)=EV(I,2)
                  IFREE= ICH(J,2)
C
C             LINK NEW MEMBER IN ORDER
                  IF(CH(J,1).LE.CH(IFIRST,1)) GO TO 8
                      GO TO 10
C                 THEN
    8                 ICH(J,1) = J
                      ICH(J,2) = IFIRST
                      ICH(IFIRST,1) = J
                      IFIRST = J
                      GO TO 90
C
```

```
   10                     K = IFIRST
   15                     K = ICH(K,2)
                          IF(CH(J,1).LT.CH(K,1)) GO TO 20
                              GO TO 25
C                         THEN LINK
   20                         ICH(J,1) = ICH(K,1)
                              ICH(J,2) = K
                              ICH(ICH(K,1),2) = J
                              ICH(K,1) = J
                              GO TO 50
C
   25                     IF(ICH(K,2).EQ.K) GO TO 30
                              GO TO 15
C                         THEN LINK ON END
   30                         ICH(K,2) = J
                              ICH(J,1) = K
                              ICH(J,2) = J
                              GO TO 50
C                    END OF LINK
C
   50         CONTINUE
C
C       IF NONE OF THE NEW EV'S WERE PLACED IN THE HULL RETURN
              IF(IFLAG.EQ.0) GO TO 90
C
C       FORM NEW HULL
              K = ICH(IFIRST,2)
C
   60         IF(ICH(K,2).EQ.K) GO TO 90
              TEST=(CH(K,2)-CH(ICH(K,1),2))*(CH(ICH(K,2),1)-CH(K,1))
     1            -(CH(ICH(K,2),2)-CH(K,2))*(CH(K,1)-CH(ICH(K,1),1))
              IF(TEST.LE.0.000) GO TO 65
                  GO TO 70
C             THEN UNLINK CH(K,-)
   65             ICH(ICH(K,1),2) = ICH(K,2)
                  ICH(ICH(K,2),1) = ICH(K,1)
                  ICH(K,2) = IFREE
                  IFREE = K
                  K = ICH(K,1)
                  IF(ICH(K,1).EQ.K) K = ICH(K,2)
                  GO TO 60
C             ELSE MOVE TO NEXT K
   70             K= ICH(K,2)
                  GO TO 60
C
   90 RETURN
      END
```

```
      SUBROUTINE  ELLIP(CH)
C***********************************************************************
C     THIS SUBROUTINE FINDS THE OPTIMAL ITERATION PARAMETERS *
C FOR THE POSITIVE HULL CH. IT RETAINS ONLY THE KEY ELEMENTS *
C IN THE HULL AND OUTPUTS THESE.                             *
C***********************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /AREA2/ D,C2,A2,FC
      COMMON /AREA3/ ICH,IFIRST,IFREE
      DIMENSION CH(25,2),ICH(25,2),ICOL(25)
C
C     BEGIN
C
C     FIRST SEE IF ANY PAIRWISE BEST IS THE OPTIMUM
          J = IFIRST
   10     IF(ICH(J,2).EQ.J) GO TO 30
              K = J
   15         IF(ICH(K,2).EQ.K) GO TO 25
                  K = ICH(K,2)
                  CALL TWOPT(CH,J,K,IFLAG)
                  IF(IFLAG.EQ.1) GO TO 15
C                 THEN TWOPT FAILED
C
C                 TEST TO SEE IF THIS PAIRWISE
C                 BEST IS OPTIMAL.
                      I = IFIRST
C
   16                 IF(I.EQ.J .OR. I.EQ.K) GO TO 18
                          GO TO 20
C                     THEN SKIP TEST
   18                     GO TO 22
C
   20                 TEST=(D-CH(I,1))*(D-CH(I,1))*(A2-C2)+
     1                    CH(I,2)*CH(I,2)*A2-A2*(A2-C2)
                      IF(TEST.GT.1.0D-8) GO TO 21
                          GO TO 22
C                     THEN TEST FAILS
   21                     GO TO 15
C
   22                 IF(ICH(I,2).EQ.I) GO TO 23
                          GO TO 24
C                     THEN TEST  WORKS
   23                     WRITE(6,500)
                          GO TO 90
C
C                     UPDATE I
   24                     I = ICH(I,2)
                          GO TO 16
```

```
C                    END OF TEST
C
   25       J = ICH(J,2)
            GO TO 10
C
C      END PAIRWISE SEARCH
C
C      FIND THE BEST THREE VALUE POINT
C
C      INITIALIZE ICOL AND TFC
   30       DO 32 I=1,25
                 ICOL(I) = 0
   32       CONTINUE
            TFC = 1.D0
C
C      TRY EACH THREE VALUE POINT
            J = IFIRST
   35       IF(ICH(ICH(J,2),2).EQ.ICH(J,2)) GO TO 70
                 K = ICH(J,2)
   40            IF (ICH(K,2).EQ.K) GO TO 65
                      L = K
   45                 IF(ICH(L,2).EQ.L) GO TO 60
                           L = ICH(L,2)
                           CALL THREPT(CH,J,K,L,IFLAG)
                           IF(IFLAG.EQ.1) GO TO 45
C                          THEN THREPT FAILED
C
C                          TEST TO SEE IF THIS THREE VALUE POINT
C                          IS A CANDIDATE
                                I = IFIRST
C
   46                           IF(I.EQ.J .OR. I.EQ.K .OR. I.EQ.L) GOTO 48
                                     GO TO 50
C                               THEN SKIP TEST
   48                                GO TO 52
C
   50                           TEST=(D-CH(I,1))*(D-CH(I,1))*(A2-C2)
     1                               +CH(I,2)*CH(I,2)*A2-A2*(A2-C2)
                                IF(TEST.GT.1.0D-8) GO TO 51
                                     GO TO 52
C                               THEN TEST FAILS
   51                                GO TO 45
C
   52                           IF(ICH(I,2).EQ.I) GO TO 53
                                     GO TO 58
C                               THEN TEST WORKS
   53                                ICOL(J) = 1
                                     ICOL(K) = 1
```

```
                                  ICCL(L) = 1
                                  IF(FC.LT.TFC) GCTO 54
                                     GOTO 56
C                                 THEN THIS POINT BEST SO FAR
   54                                TFC = FC
                                     TC = D
                                     TC2 = C2
                                     TA2 = A2
   56                             GC TO 45
C
C                              UPDATE I
   58                             I = ICH(I,2)
                                  GC TC 46
C                       END OF TEST
C
   6C            K = ICH(K,2)
                 GC TO 40
   65         J = ICH(J,2)
              GO TO 35
C        END OF SEARCH
C
C        THE BEST THREE VALUE PCINT FIT HAS BEEN FCUND
   70         C = TD
              C2 = TC2
              A2 = TA2
              FC = TFC
C
C        SAVE ONLY KEY ELEMENTS
              I = IFIRST
   8C         K = ICH(I,2)
              IF(ICOL(I).EQ.0) GO TO 82
                 GO TO 84
C             THEN UNLINK CH(I,-)
   82            ICH(ICH(I,1),2) = ICH(I,2)
                 ICH(ICH(I,2),1) = ICH(I,1)
                 ICH(I,2) = IFREE
                 IFREE = I
   84         IF(K.EQ.I) GC TC 90
              I = K
              GC TO 80
C
C        OUTPUT NEW PARAMETERS AND POSITIVE HULL
   90         WRITE(6,600)
              WRITE(6,700) D,C2,A2,FC
              WRITE(6,800)
              I = IFIRST
   95            WRITE(6,900) CH(I,1),CH(I,2)
                 IF(ICH(I,2).EQ.I) GC TO 99
```

```
            I = ICH(I,2)
            GO TO 95
C
C     FORMATS:
  500     FORMAT(' THE OPTIMAL IS A PAIRWISE BEST.')
  600     FORMAT(' THE NEW PARAMETERS ARE:')
  700     FORMAT(' C = ',D12.5,3X,'C2 = ',D12.5,3X,'A2 = ',
     1        D12.5,3X,'FC = ',D12.5)
  800     FORMAT(' THE POSITIVE HULL:')
  900     FORMAT(' ',3X,D12.5,' +I*',D12.5)
C
   99 RETURN
      END
```

```
      SUBROUTINE TWOPT(CH,J,K,IFLAG)
C*****************************************************************
C     THIS SUBROUTINE FINDS THE OPTIMUM ITERATION PARAMETERS *
C FOR THE TWO EIGENVALUES: CH(J,-) AND CH(K,-)               *
C*****************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /AREA2/ D,C2,A2,FC
      DIMENSION CH(25,2)
C
C     SET THE CONSTANTS
          A = (CH(K,1)-CH(J,1))/2.0D0
          B = (CH(K,1)+CH(J,1))/2.0D0
          S = (CH(K,2)-CH(J,2))/2.0D0
          T = (CH(K,2)+CH(J,2))/2.0D0
C
C     ELIMINATE DEGENERATE CASES
C     CASE I: T SMALL
          IF(T/B.LT.1.0D-3) GO TO 5
              GO TO 10
C         THEN
    5         D = B
              C2 = A*A
              A2 = C2
              FC = A/(D+DSQRT(D*D-C2))
              GO TO 50
C
C     CASE II: A SMALL
   10     IF(A/T.LT.1.0D-3) GO TO 15
              GO TO 20
C         THEN
   15         D = B
              Y = T+DABS(S)
              C2 = -Y*Y
              A2 = A*(A+DSQRT(A*A+4.D0*Y*Y))/2.0D0
              FC = (DSQRT(A2)+DSQRT(A2-C2))/(D+DSQRT(D*D-C2))
              GO TO 50
C
C     CASE III: S SMALL
   20     IF(DABS(S)/A.LT.1.0D-3) GO TO 25
              GO TO 30
C         THEN
   25         CALL ACUBIC(A,B,T)
              GO TO 50
C
C     CASE IV: GENERAL CASE
   30     CALL FIFTH(A,B,S,T,IFLAG)
C
   50 RETURN
      END
```

```
      SUBROUTINE THREPT(CH,J,K,L,IFLAG)
C*****************************************************************
C     THIS SUBROUTINE FINDS THE ITERATION PARAMETERS             *
C ASSOCIATED WITH THE UNIQUE ELLIPSE THROUGH THE THREE           *
C EIGENVALUES: CH(J,-),CH(K,-), AND CH(L,-)                      *
C*****************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /AREA2/ D,C2,A2,FC
      DIMENSION CH(25,2)
C
C     SET IFLAG
         IFLAG = 0
C
C     COMPUTE D
         W = CH(J,2)*CH(J,2)*(CH(K,1)*CH(K,1)-CH(L,1)*CH(L,1))
     1        +CH(L,2)*CH(L,2)*(CH(J,1)*CH(J,1)-CH(K,1)*CH(K,1))
     2        +CH(K,2)*CH(K,2)*(CH(L,1)*CH(L,1)-CH(J,1)*CH(J,1))
C
         Z = CH(J,2)*CH(J,2)*(CH(K,1)-CH(L,1))
     1        +CH(L,2)*CH(L,2)*(CH(J,1)-CH(K,1))
     2        +CH(K,2)*CH(K,2)*(CH(L,1)-CH(J,1))
C
         IF(Z.LE.0.0D0) GO TO 10
            GO TO 20
C        THEN NO ELLIPSE FITS THESE THREE VALUES
   10       IFLAG = 1
            GO TO 90
C
   20    D = W/(2.0D0*Z)
C
         IF(D.LE.0.0D0) GO TO 30
            GO TO 40
C        THEN ELLIPSE CONTAINS THE ORIGIN
   30       IFLAG = 1
            GO TO 90
C
C     COMPUTE OTHER ITERATION PARAMETERS
   40    A2=(CH(J,2)*CH(J,2)*CH(L,1)*CH(K,1)*(CH(L,1)-CH(K,1))
     1     +CH(L,2)*CH(L,2)*CH(J,1)*CH(K,1)*(CH(K,1)-CH(J,1))
     2     +CH(K,2)*CH(K,2)*CH(J,1)*CH(L,1)*(CH(J,1)-CH(L,1)))
     3     /Z + D*D
C
         IF(A2.GE.D*D) GO TO 50
            GO TO 60
C        THEN ELIPSE CONTAINS THE ORIGIN
   50       IFLAG = 1
            GO TO 90
C
   60    C2=A2*(1.D0-Z/(CH(J,1)*CH(L,1)*(CH(J,1)-CH(L,1))
     1     +CH(L,1)*CH(K,1)*(CH(L,1)-CH(K,1))
     2     +CH(J,1)*CH(K,1)*(CH(K,1)-CH(J,1))))
C
         FC=(DSQRT(A2)+DSQRT(A2-C2))/(D+DSQRT(D*D-C2))
C
   90 RETURN
      END
```

```
      SUBROUTINE ACUBIC(A,B,T)
C*****************************************************************
C     THIS SUBROUTINE FINDS THE BEST ITERATION PARAMETERS        *
C WHEN BOTH EIGENVALUES HAVE THE SAME IMAGINARY PART. THE        *
C BEST VALUE OF A2 IS FOUND AS THE ROOT OF A CUBIC POLY-         *
C NOMIAL.                                                        *
C*****************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /AREA2/ D,C2,A2,FC
C
C     TO COMPUTE D,A2,C2,FC
          D=B
          X=(B*B*A*A*A*A*T*T)*((B*B+T*T)*(B*B+T*T)+A*A*
     1        (T*T-B*B))/(2.D0*(B*B+T*T)*(B*B+T*T)*(B*B+T*T))
          Z=(B*B*A*A*A*A*T*T)/((B*B+T*T)*(B*B+T*T))
          Y1=X+DSQRT(X*X+Z*Z*Z)
          Y2=X-DSQRT(X*X+Z*Z*Z)
          Y=DSIGN(DABS(Y1)**(1.D0/3.D0),Y1)+DSIGN(DABS(Y2)**
     1        (1.D0/3.D0),Y2)
          A2=Y+(B*B*A*A)/(B*B+T*T)
          C2=(A2*(A2-T*T-A*A))/(A2-A*A)
           FC = (DSQRT(A2)+DSQRT(A2-C2))/(D+DSQRT(D*D-C2))
C
      RETURN
      END
```

```
      SUBROUTINE FIFTH(A,B,S,T,IFLAG)
C***********************************************************************
C     THIS SUBROUTINE FINDS THE OPTIMAL ITERATION PARAMETERS *
C FOR TWO COMPLEX EIGENVALUES. THE BEST VALUE OF D IS FOUND  *
C AS THE ROOT OF A FIFTH DEGREE POLYNOMIAL WITH COEFICIENTS  *
C           P1,P2,P3,P4,P5,P6. (COMMON /AREA4/).             *
C***********************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /AREA2/ D,C2,A2,FC
      COMMON /AREA4/ P1,P2,P3,P4,P5,P6
C
C     COMPUTE THE FIFTH DEGREE POLYNOMIAL
         R1 = -A*T/S
         R2 = -A*S/T
         R3 =  S*T/A
         R4 = -B
C
         P1=R1*(R1+2.D0*R2+R3-4.D0*R4)+R2*(R2+R3-4.D0*R4)
     1         +R4*(4.D0*R4-2.D0*R3)
         P2=R1*(4.D0*R4*(R3-R4)+R2*(12.D0*R4-5.D0*R3-4.D0*R2)
     1         +R1*(2.D0*R4-R3-4.D0*R2))
     2         +R2*(4.D0*R4*(R3-R4)+R2*(2.D0*R4-R3))
     3         -R3*R4*R4
         P3=R1*(R2*(R4*(2.D0*R4-4.D0*R3)+R2*(3.D0*R3-4.D0*R4))
     1         +R1*(R4*R4+R2*(4.D0*R2+3.D0*R3-4.D0*R4)-R1*R3))
     2         +R2*R2*(R4*R4-R2*R3)
         P4=R1*R2*R3*(R1*(3.D0*R1-4.D0*R4)+R2*(3.D0*R2-4.D0*R4)
     1         +2.D0*R4*R4)
         P5=3.D0*R1*R1*R2*R2*R3*(2.D0*R4-R1-R2)
         P6=R1*R1*R2*R2*R3*(R1*R2-R4*R4)
C
C     COMPUTE Y AND Z
         Y = (T-S)*(T-S)*(B+A)*(B+A)-(T+S)*(T+S)*(B-A)*(B-A)
         Z = (T-S)*(T-S)*(B+A) - (T+S)*(T+S)*(B-A)
C
      IF(S.LT.0.D0) GO TO 10
         GO TO 20
C     THEN
   10    E1 = DMAX1(-A,(Y/(2.D0*Z)-B))
         E2 = 0.D0
         GO TO 50
C     ELSE
   20    IF(Y.GT.2.D0*B*Z) GO TO 30
            GO TO 40
C        THEN
   30       E1 = 0.D0
            E2 = A
            GO TO 50
```

```
C          ELSE
   40          E1 = 0.C0
               E2 = DMIN1(A,Y/(2.C0*Z)-B)
               GO TO 50
C
C       FIND ROOT
C  ***ANY LIBRARY ROUTINE MAY BE USED TC FIND THE REAL ROOT***
C  ***  CF THE FIFTH DEGREE PCLY IN THE INTERVAL (E1,E2).   ***
C  ***            THE ROOT SHCULC BE RETURNEC AS E1.         ***
C  ***            ZEROIN APPEARS IN APPENDIX C.              ***
C
   50      CALL ZERCIN(E1,E2,IFLAG)
C
           IF(IFLAG.GE.3) GC TC 60
             GO TC 70
C          THEN ZEROIN FAILEC
   60          IFLAG = 1
               GC TC 80
C          ELSE
   70          IFLAG = 0
C
C       CCMPUTE ITERATION PARAMETERS
           D = E1 + B
           A2= (E1-R1)*(E1-R2)
           C2= A2*(E1-R3)/(E1)
           FC= (DSORT(A2)+DSCRT(A2-C2))/(D+DSCRT(D*D-C2))
C
   80 RETURN
      END
```

APPENDIX B

```
      SUBROUTINE BIDIAG(A,X,B)
C *****************************************************************
C     THIS SUBROUTINE FINDS THE LEAST SQUARE SOLUTION OF THE   *
C SYSTEM: A(N,M)*X(M) - B(N) = 0.                              *
C *****************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(4,4), X(4),E(4),U(4,6),V(4,6),VW(4),T1(4),T2(4)
      DIMENSION P(6)
      N = 4
      M = 4
      IMAX = 4
C
C     BEGIN
C
C     INITIALIZE ISTEP
          ISTEP = 0
C
C     INITIALIZE VECTORS AND CONSTANTS
C         SET X(M),VW(M) = 0
               DO 2 I=1,M
                    X(I) = 0.D0
                    VW(I) = 0.D0
    2          CONTINUE
C
C         SET B1*U(N) = B(N)
               CALL INPRO(B,B,B1,N)
               B1 = DSQRT(B1)
               DO 6 I=1,N
                    U(I,1) = B(I)/B1
    6          CONTINUE
C
C         INITIALIZE A1,W,Z,T
               A1 = 0.D0
               W = 0.D0
               Z = -1.D0
               T = 1.D0/B1
C
C     SET EBND
          EBND = B1*1.D0-10
C
C     FIRST STEP: COMPUTE FIRST V AND A1
          CALL TMUL(V(1,1),U(1,1),A,N,M)
          CALL INPRO(V(1,1),V(1,1),A1,M)
          A1 = DSQRT(A1)
            DO 8 I=1,M
               V(I,1) = V(I,1)/A1
    8       CONTINUE
C
```

```
C       MAIN LOOP
C            UPDATE ISTEP
   10              ISTEP = ISTEP+1
C
C            COMPUTE NEW X
                   Z = -Z*B1/A1
                   DO 12 I=1,M
                         X(I) = X(I) + Z*V(I,ISTEP)
   12              CONTINUE
C
C            COMPUTE NEW VW
                   W = (T-B1*W)/A1
                   DO 14 I=1,M
                         VW(I) = VW(I) + W*V(I,ISTEP)
   14              CONTINUE
C
C            COMPUTE NEW U AND B1
                   CALL AMUL(T1,V(1,ISTEP),A,N,M)
                   DO 16 I=1,N
                         U(I,ISTEP+1) = T1(I) - A1*U(I,ISTEP)
   16              CONTINUE
                   CALL INPRO(U(1,ISTEP+1),U(1,ISTEP+1),B1,N)
                   B1 = DSQRT(B1)
C
C            TEST B1
                   IF(B1.LT.EBND) GO TO 18
                         GO TO 20
C                  THEN SOLUTION IS EXACTLY X
   18                    GO TO 100
C                  ELSE RE-ORTHOGONALIZE
   20                    K = ISTEP+1
                         DO 22 I=1,M
                               U(I,K) = U(I,K)/B1
   22                    CONTINUE
                         DO 24 I=1,ISTEP
                            CALL INPRO(U(1,K),U(1,I),P(I),M)
   24                    CONTINUE
                         DO 26 I=1,ISTEP
                            DO 26 J=1,M
                              U(J,K)=U(J,K)-P(I)*U(J,I)
   26                    CONTINUE
                         CALL INPRO(U(1,K),U(1,K),B2,M)
                         B2 = DSQRT(B2)
                         DO 28 I=1,M
                              U(I,K) = U(I,K)/B2
   28                    CONTINUE
C
C            UPDATE T
```

```
                T = -T*A1/B1
C
C         TEST ISTEP
                IF(ISTEP.GE.IMAX) GO TO 100
C
C         COMPUTE NEW V AND A1
                CALL TMUL(T2,U(1,ISTEP+1),A,N,M)
                DO 30 I=1,M
                    V(I,ISTEP+1) = T2(I) - B1*V(I,ISTEP)
   30           CONTINUE
                CALL INPRO(V(1,ISTEP+1),V(1,ISTEP+1),A1,M)
                A1 = DSQRT(A1)
C
C         TEST A1
                IF(A1.LT.EBND .OR. ISTEP.EQ.4) GO TO 32
                    GO TO 38
C               THEN SOLUTION IS X=X-G*VW
   32               G = B1*Z/(B1*W-T)
                    DO 34 I=1,M
                       X(I) = X(I) - G*VW(I)
   34               CONTINUE
                    GO TO 100
C               ELSE RE-ORTHOGONALIZE
   38               K = ISTEP+1
                    DO 40 I=1,M
                       V(I,K) = V(I,K)/A1
   40               CONTINUE
                    DO 42 I=1,ISTEP
                      CALL INPRO(V(1,K),V(1,I),P(I),M)
   42               CONTINUE
                    DO 44 I=1,ISTEP
                      DO 44 J=1,M
                        V(J,K)=V(J,K)-P(I)*V(J,I)
   44               CONTINUE
                    CALL INPRO(V(1,K),V(1,K),A2,M)
                    A2 = DSQRT(A2)
                    DO 46 I=1,M
                       V(I,K) = V(I,K)/A2
   46               CONTINUE
C
C         REPEAT LOOP
                GO TO 10
C      END OF LOOP
C
  100 RETURN
      END
```

```
      SUBROUTINE AMUL(Y,X,A,N,M)
C ********************************************************************
C     THIS SUBROUTINE DOES THE MATRIX MULTIPLICATION:                 *
C             Y(N) = A(N,M)*X(M).                                     *
C ********************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(N,M),X(M),Y(N)
C
C     BEGIN
         DO 20 I=1,N
            Y(I) = 0.D0
            DO 10 K=1,M
               Y(I) = Y(I) + A(I,K)*X(K)
   10       CONTINUE
   20    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TMUL(Y,X,A,N,M)
C ********************************************************************
C     THIS SUBROUTINE DOES THE MATREX MULTIPLICATION:                 *
C             Y(M) = AT(M,N)*X(N),                                    *
C     WHERE AT IS THE TRANSPOSE OF A.                                 *
C ********************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(N,M),X(N),Y(M)
C
C     BEGIN
         DO 20 K=1,M
            Y(K) = 0.D0
            DO 10 I=1,N
               Y(K) = Y(K) + A(I,K)*X(I)
   10       CONTINUE
   20    CONTINUE
      RETURN
      END
```

APPENDIX C

```
      SUBROUTINE RSSR (A,     ROOTS,DEGREE,M,MMAX,DELTA,EPSIL   ,D)
      DIMENSION A(51),   IA(51,3),ROOTS(2,50),D(51),RCMOD(50),
     1NONRT(25),MNONRT(25)
      REAL*8       A,ROOTS,D,RCMOD,DELTA,EPSIL
            INTEGER DEGREE
      NCUR=DEGREE
    7 NL=NCUR
      CALL ROOTSQ (A,IA,NCUR,M)
      CALL REALRT    (A,IA,M,NCUR,DELTA,EPSIL   ,RCMOD,IA,
     1NONRT,MNONRT,NCO,ROOTS)
      IF (NCO.EQ.0) GO TO 12
      CALL COMPRT    (A,IA,RCMOD,ROOTS,M,MNONRT,NONRT,
     1         NCO,DELTA,EPSIL   ,NCUR)
      IF (NCUR.EQ.0) GO TO 12
      IF (NCUR.EQ.NL) M=M+1
      IF (M.LE.MMAX) GO TO 7
      GO TO 13
   12 CALL RECON (ROOTS,A,                 D,DEGREE)
   13 DEGREE=NCUR
      RETURN
      END



      SUBROUTINE ROOTSQ (A,IA,NCUR,MM)
      DIMENSION A(51,3),IA(51,3)
      REAL*8       A,X
      N1=NCUR+1
      DO 1 J=1,N1
      A(J,3)=A(J,1)
      IA(J,3)=0
      CALL SCAL (A(J,3),IA(J,3))
    1 CONTINUE
      DO 9 M=1,MM
      DO 2 J=1,N1
      A(J,2)=A(J,3)
      IA(J,2)=IA(J,3)
      A(J,3)=0.0
    2 CONTINUE
      DO 6 J=1,N1
      KM= MINO (N1-J,J-1)
      IF (KM.EQ.0) GO TO 5
      DO 4 L=1,KM
      JL=J-L
      JLP=J+L
      X=A(JL,2)*A(JLP,2)
      LR= MOD (L,2)
      IF (LR.EQ.1) X=-X
      IX=IA(JL,2)+IA(JLP,2)
      CALL ADD (A(J,3),IA(J,3),X,IX)
    4 CONTINUE
```

```
      A(J,3)=2.0*A(J,3)
    5 X=A(J,2)**2
      IX=IA(J,2)+IA(J,2)
      CALL ADD (A(J,3),IA(J,3),X,IX)
      JR= MOD  (J,2)
      IF (JR.EQ.0) A(J,3)=-A(J,3)
    6 CONTINUE
    9 CONTINUE
      RETURN
      END



      SUBROUTINE REALRT    (A,IA,M,NCUR,DELTA,EPSIL   ,ROMOD,MROMOD,
     1NONRT,MNONRT,NCO,ROOTS)
      DIMENSION A(51,3),IA(51,3),ROOTS(2,50),ROMOD(50),MROMOD(50),
     1NONRT(25),MNONRT(25),           IPIV(51)
      REAL*8       A,ROOTS,ROMOD,T,XN,W,         Q,DELTA,EPSIL
      NCUR1=NCUR+1
      DO 1 I=1,NCUR1
    1 IPIV(I)=1
      IF (NCUR.EQ.1) GO TO 8
      DO 7 I=2,NCUR
      IF (A(I,3).EQ.0.0) GO TO 6
      T=      A(I,2)**2/A(I,3)
      IF (MOD(I,2).EQ.0) T=-T
      IT=IA(I,2)+IA(I,2)-IA(I,3)
      XN=T*64.0**IT
      IF(DABS(1.-XN).LT.DELTA) GO TO 7
    6 IPIV(I)=0
    7 CONTINUE
    8 I=1
      I4=0
      DO 12 I2=2,NCUR1
      IF (IPIV(I2).EQ.0) GO TO 12
      I4=I4+1
      I1=I2-I
      ROMOD(I4)=DABS(A(I2,3)/A(I,3))
      IROMOD=IA(I2,3)-IA(I,3)
      IF (ROMOD(I4).EQ.0.0) GO TO 11
      T=2.0**M*FLOAT(I1)
      XN=(DLOG(ROMOD(I4))+DFLOAT(IROMOD)*4.158883083359671860O)/T
      ROMOD(I4)=DEXP(XN)
   11 MROMOD(I4)=I1
      I=I2
   12 CONTINUE
      Q=0.0
      NCO=0
      DO 22 I=1,I4
      KL=I4+1-I
      W=-ROMOD(KL)
      I5=MROMOD(KL)
      DO 20 J=1,I5
```

```
14 CALL TEST (A,    W,Q,NCUR,RCMOD(KL),EPSIL   ,K)
   IF (K.EQ.0) GC TC 21
   ROOTS(1,NCUR)=-W
   ROOTS(2,NCUR)=0.0
   DO 16 L=2,NCUR1
16 A(L,1)=A(L,1)-A(L-1,1)*W
20 NCUR=NCUR-1
   GO TC 22
21 W=-W
   IF (W.GT.0.0) GO TC 14
   NCO=NCC+1
   NONFT(NCO)=KL
   MNONPT(NCO)=I5+1-J
22 CONTINUE
   RETURN
   END
```

```
      SUBROUTINE COMPRT   (A,IA,ROMOD,ROOTS,M,MNCNRT,NONRT,
     1        NCO,DELTA,EPSIL   ,NCUR)
      DIMENSION A(51),IA(51),ROMOD(50),ROOTS(2,50),SR(48,3),ISR(48,3),
     1SROMOD(47),SROOTS(2,47),MNCNRT(25),NONRT(25),NSONRT(23),MSNORT(23)
      REAL*8      A,ROMOD,ROOTS,SR,SROMOD,SROOTS,       W,Q,DELTA,EPSIL
      DO 23 I=1,NCO
      JA=NCNRT(I)
      I1=MNONRT(I)/2
      DO 22 J=1,I1
      CALL SUBRES (A,IA,NCUR,SR,ISR,ROMOD(JA))
      NSCUR=NCUR-3+4/NCUR
      J2=NSCUR
      IF (NSCUR.NE.1) GO TO 9
      SROOTS(1,1)=-SR(2,1)/SR(1,1)
      NSCUR=0
      GO TO 11
    9 CALL ROOTSQ (SR,ISR,NSCUR,M)
      CALL REALRT   (SR,ISR,M,NSCUR,DELTA,EPSIL   ,SROMOD, ISR,
     1NSONRT,MSNORT,NSCC,SROOTS)
   11 LL=J2-NSCUR
      L=1
   12 L2=J2+1-L
      IF (LL.EQ.0) SROOTS(1,L2)=0.0
      IF(DABS(SROOTS(1,L2)).GE.2.0) GO TO 16
      W=SROOTS(1,L2)*ROMOD(JA)
      Q =ROMOD(JA)*ROMOD(JA)
      CALL TEST (A,   W,Q ,NCUR,ROMOD(JA),EPSIL   ,K)
      IF (K.EQ.0) GO TO 16
      ROOTS(1,NCUR)=-W/2.0
      ROOTS(2,NCUR)=DSQRT(Q-(W/2.0)**2)
      ROOTS(1,NCUR-1)=ROOTS(1,NCUR)
      ROOTS(2,NCUR-1)=-ROOTS(2,NCUR)
      CALL QUADIV (NCUR,A,   W,Q)
      NCUR=NCUR-2
      GO TO 22
   16 L=L+1
      IF (L.LE.LL) GO TO 12
   22 CONTINUE
   23 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TEST (A,    W,Q,N,ROMOD,EPSIL  ,K)
      DIMENSION A(51),            B(3),     T(2),E(2)
      REAL*8      A,B,W,                E,  DIF,Q,ROMOD,T,EPSIL
      B(2)=0.0
      B(3)=A(1)
      DO 2 I=1,N
      B(1)=B(2)
      B(2)=B(3)
      B(3)=A(I+1)-W*B(2)-Q*B(1)
    2 CONTINUE
      T(1)=0.0
      T(2)=0.0
      N1=N+1
      E(1)=ROMOD+2.0*EPSIL*ROMOD
      E(2)=ROMOD+EPSIL*ROMOD
      DO 7 I=1,N1


      T(1)=T(1)*E(1)+DABS(A(I))
      T(2)=T(2)*E(2)+DABS(A(I))
    7 CONTINUE
      DIF=T(1)-T(2)
      K=0
      IF (Q.NE.0.0) GO TO 18
      IF(DIF.GE.DABS(B(3))) K=1
      RETURN
   18 IF (DIF**2.GE.(Q*B(2)**2+W*B(2)*B(3)+B(3)**2)) K=1
      RETURN
      END
```

```
      SUBROUTINE SUBRES (A,IA,N,SR,ISR,RCMCD)
      DIMENSION A(51),  IA(51),  SR(48),  ISR(48),  C(51),B(48,3)
      REAL*8      A,SR,RCMCD,  T,C,  B
      N1=N+1
      T=1.0
      DO 1 I=1,N1
      J=N1-I+1
      C(J)=A(J)  *T
      T=T*ROMOD
    1 CONTINUE
      IF (N.LE.4) GO TO 16
      N2=N-2
      DO 3 I=1,N2
      B(I,1)=0.0
    3 CONTINUE
      B(1,2)=C(1)
      DO 8 I=2,N2
      B(1,3)=C(I)-B(1,1)
      DO 5 J=2,I
      B(J,3)=-B(J-1,2)-B(J,1)
    5 CONTINUE
      DO 7 J=1,I
      IF (J.NE.I) B(J,1)=B(J,2)
      B(J,2)=B(J,3)
    7 CONTINUE
    8 CONTINUE
      SR(N2)  =C(N)-B(1,3)
      SR(N2-1)  =-C(N1)-B(2,3)
      DO 10 J=3,N2
      K=N2+1-J
      SR(K)  =-B(J,3)
   10 CONTINUE
      RETURN
   16 SR(1)  =C(1)
      SR(2)  =-C(2)
      IF (N.EQ.2) RETURN
      SR(3)  =C(3)-C(1)-C(5)
      RETURN
      END
```

```
      SUBROUTINE RECON (ROOTS,A,    D,N)
      DIMENSION ROOTS(2,50),D(51)
      REAL*8       ROOTS,A,D,  T,U
      DO 1 I=1,N
      D(I)=0.0
    1 CONTINUE
      D(N+1)=1.0
      NS=N+1
      DO 10 I=1,N
      NL=NS-I
      IF (ROOTS(2,I)) 3,7,10
    3 T=ROOTS(1,I)**2+ROOTS(2,I)**2
      U=2.0*ROOTS(1,I)
      DO 5 J=NL,N
      D(J-1)=T*D(J-1)-U*D(J)+D(J+1)
    5 CONTINUE
      D(N)=T*D(N)-U*D(N+1)
      GO TO 9
    7 T=-ROOTS(1,I)
      DO 8 J=NL,N
      D(J)=T*D(J)+D(J+1)
    8 CONTINUE
    9 D(N+1)=T*D(N+1)
   10 CONTINUE
      DO 11 II=1,NS
      D(II)=D(II)*A
   11 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE QUADIV (N,A,    W,Q)
      DIMENSION A(51)
      REAL*8       A,  W,Q,XN
      N1=N+1
      DO 3 I=2,N1
      XN=A(I-1)*W
      IF (I.NE.2) XN=XN+A(I-2)*Q
      A(I)=A(I)-XN
    3 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ADD (X,IX,Y,IY)
      REAL*8       X,Y,    B
      IF (X.NE.0.0) GO TO 3
      X=Y
      IX=IY
      GO TO 13
    3 IF (Y.EQ.0.0) RETURN
      B=Y
      IDIFF=IX-IY
      IF (IDIFF) 5,12,8
    5 B=X
      X=Y
      IX=IY
      IDIFF=-IDIFF
    8 IF (IDIFF.GE.14) GO TO 13
      DO 11 I=1,IDIFF
      B=B/64.0
   11 CONTINUE
   12 X=X+B
   13 CALL SCAL  (X,IX)
      RETURN
      END


      SUBROUTINE SCAL  (X,IX)
      REAL*8       X,Y
      Y=DABS(X)
      IF (Y.NE.0.0) GO TO 3
      IX=0
      RETURN
    3 IF (Y.LE.64.0) GO TO 5
      Y=Y/64.0
      IX=IX+1
      GO TO 3
    5 IF (Y.GE.0.015625) GO TO 7
      Y=Y*64.0
      IX=IX-1
      GO TO 5
    7 IF (X.LT.0.0) Y=-Y
      X=Y
      RETURN
      END
```

```
      SUBROUTINE ZEROIN(B,C,IFLAG)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /AREA4/ P1,P2,P3,P4,P5,P6
      F(Z)= ((((P1*Z+P2)*Z+P3)*Z+P4)*Z+P5)*Z+P6
      ABSERR = 0.D0
      RELERR = 0.D0
      U=9.0D-15
      RE=DMAX1(RELERR,U)
      IC=0
      ACBS=DABS(B-C)
      A=C
      FA=F(A)
      FB=F(B)
      FC=FA
      KOUNT=2
      FX=DMAX1(DABS(FB),DABS(FC))
1     IF(DABS(FC).GE.DABS(FB))GO TO 2
      A=B
      FA=FB
      B=C
      FB=FC
      C=A
      FC=FA
2     CMB=0.5D0*(C-B)
      ACMB=DABS(CMB)
      TOL=RE*DABS(B)+ABSERR
      IF(ACMB.LE.TOL)GO TO 8
      IF(KOUNT.GE.500)GO TO 12
      P=(B-A)*FB
      Q=FA-FB
      IF(P.GE.0.D0)GO TO 3
      P=-P
      Q=-Q
3     A=B
      FA=FB
      IC=IC+1
      IF(IC.LT.4)GO TO 4
      IF(8.0D0*ACMB.GE.ACBS)GO TO 6
      IC=0
      ACBS=ACMB
4     IF(P.GT.DABS(Q)*TOL)GO TO 5
      B=B+DSIGN(TOL,CMB)
      GO TO 7
5     IF(P.GE.CMB*Q)GO TO 6
      B=B+P/Q
      GO TO 7
```

```
 6    B=0.5D0*(C+B)
 7    FB=F(B)
      IF(FB.EQ.0.D0)GO TO 9
      KOUNT=KOUNT+1
      IF(DSIGN(1.0D0,FB).NE.DSIGN(1.0D0,FC))GO TO 1
      C=A
      FC=FA
      GO TO 1
 8    IF(DSIGN(1.0D0,FB).EQ.DSIGN(1.0D0,FC))GO TO 11
      IF(DABS(FB).GT.FX)GO TO 10
      IFLAG=1
      RETURN
 9    IFLAG=2
      RETURN
10    IFLAG=3

      RETURN
11    IFLAG=4
      RETURN
12    IFLAG=5
      RETURN
      END
```

VITA

The author was born November 15, 1948, in Woodstock, Illinois. After spending the majority of his childhood in Crystal Lake, Illinois, he attended the University of Wisconsin in Madison, Wisconsin, where he received a B.S. in Mathematics in June of 1970. He started graduate work at the University of Illinois in the fall of 1970, but withdrew for military duty. After this return to the University of Illinois in early 1971, he received a M.S. in Mathematics in January of 1972 and a Ph.D. in October of 1975. In the fall of 1975 he joined the faculty at Emory University in Atlanta, Georgia.

BIBLIOGRAPHIC DATA SHEET	1. Report No. UIUCDCS-R-75-758	2.	3. Recipient's Accession No.

Title and Subtitle
AN ITERATIVE METHOD FOR SOLVING NONSYMMETRIC LINEAR SYSTEMS WITH DYNAMIC ESTIMATION OF PARAMETERS

5. Report Date: October 1975

6.

Author(s) Thomas Albert Manteuffel

8. Performing Organization Rept. No.

Performing Organization Name and Address

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

10. Project/Task/Work Unit No.

11. Contract/Grant No.
NSF GJ-36393
DCR74-23679 (NSF)

12. Sponsoring Organization Name and Address

National Science Foundation
Washington, D.C.

13. Type of Report & Period Covered

14.

15. Supplementary Notes

16. Abstracts

The subject of this thesis is an iterative method for solving large, sparse linear systems, based upon the scaled and translated Tchebychef polynomials. It is shown that such an iteration is optimal, in a certain sense, over all polynomial based gradient methods.

Further, an algorithm is developed by which the best scaling and translating factors may be found from knowledge of the eigenvalues of the linear system. Since the eigenvalues of the linear system are seldom known, a dynamic procedure is developed to improve the choice of scaling and translating parameters from knowledge obtained during iteration.

Finally, experimental results are described which show this method to be of potential value on a large class of problems, especially in conjunction with factorization methods.

17. Key Words and Document Analysis. 17a. Descriptors

Linear
Iterative
Nonsymmetric
Dynamic
Tchebychef

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages
	20. Security Class (This Page) UNCLASSIFIED	22. Price